6-14-2018

At-Home Neurofeedback Treatment

Sanah Imran
Santa Clara University, simran@scu.edu

Frank Cannizzaro
Santa Clara University, fcannizzaro@scu.edu

SANTA CLARA UNIVERSITY

Department of Electrical Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Frank Cannizzaro, Sanah Imran

ENTITLED

AT-HOME NEUROFEEDBACK TREATMENT

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE
IN
ELECTRICAL ENGINEERING

Sloba Krishnan

Thesis Advisor

Sloba Krishnan

Department Chair

AT-HOME NEUROFEEDBACK TREATMENT

By

Sanah Imran, Frank Cannizzaro

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical Engineering

Santa Clara, California

June 14, 2018

At-Home Neurofeedback Treatment

Sanah Imran and Frank Cannizzaro

Department of Electrical Engineering
Santa Clara University
June 14, 2018

Abstract. In the past two decades, the prevalence of mental health illnesses and cognitive disorders has dramatically increased both in the United States and worldwide. While the problem is an expensive one, costing billions annually in lost productivity, the greatest cost is to the patient suffering from mental health problems. Those who have the means to seek professional help may be prescribed medications that often have negative side effects and merely mitigate symptoms of these disorders without treating the underlying cause. Fortunately, there is a new and revolutionary therapy available called neurofeedback. It uses electroencephalogram (EEG) signals to self-regulate brain function; however, it is a costly treatment that requires visits to a clinician's office. For our project, we designed a device that allows for lower cost, home-based neurofeedback treatment.

Keywords: biofeedback, neurofeedback, EEG.

Acknowledgements

This senior design project could not have been possible without the time of effort of several individuals. We would like to thank our advisor, Dr. Shoba Krishnan, for her experience and guidance to keep this project on track. We would also like to recognize our TA, Nick Mikstas, for his assistance in both filter design and selecting ICs. To Dr. Sarah Wilson, we appreciate all the drop-in appointments on digital signal processing. To the Electrical Engineering lab manager, Yohannes Kahsai, thank you for your constant supervision and helping us find any materials we required. To Chang Ong, we are grateful for your assistance designing the box for our final prototype. Finally, to Mir Imran, without your wealth of knowledge and expertise, this project may not have been possible.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background on Neurofeedback

The digital age has brought much advancement in healthcare. Life expectancy has increased and the mortality rates of many health conditions, from diabetes to heart disease, have decreased. However, mental health remains stigmatized and largely invisible in the public understanding of health, even though millions of Americans suffer from these issues. And these diseases are on the rise, creating a need for innovative treatments [1].

Currently, one in five American adults experiences a mental illness in any given year [2]. Mental illnesses are also common among young people, affecting 21% of people between the ages of 13 and 18 years [2]. Not only are these illnesses debilitating for those who suffer from them, but they also contribute to massive economic costs. In the United States alone, mental illnesses cost $193 billion per year in lost productivity [3].

Because mental health issues are not very well understood, the primary treatment options often do not address all the underlying causes of the disease. A study by the NIH found that of the 16 million Americans with diagnosed clinical depression, half use antidepressants to treat their illnesses [4]. While antidepressants may help some people recover the ability to function normally, many also come with a host of negative side effects. About two-thirds of people who suffer from depression do not respond well to drugs or psychological intervention [5]. Titration is another issue with antidepressants, as the correct dosage and drug must be prescribed for the patient to reap its intended benefits. Recently, many clinicians and psychiatrists have also made stronger arguments against the over-prescription of such medications. Dr. Mark Olfson, a professor of clinical psychiatry at Columbia, has written, “many people with mild depression are prescribed antidepressants even though they aren't likely to benefit from the drugs” [6].

However, alternatives to medications do exist. A recent therapy for mood and mental health disorders is neurofeedback, and experimental results show high promise for its efficacy in treating these illnesses [7].

Neurofeedback is a therapy that promotes self-regulation of brain activity. A patient undergoing the treatment wears a device that monitors electroencephalogram (EEG) signals produced by the brain (known colloquially as "brainwaves"). These signals contain various frequencies, and each frequency is commonly associated with a particular mental state. By analyzing the frequency components of the signal, the system can predict the mental state of the patient. The feedback portion of the treatment involves an audiovisual component that changes in response to this mental state.

For example, if a child is undergoing neurofeedback for ADHD treatment, the audiovisual component could be an airplane-flying game. If the system indicates a state of attention, the airplane will continue to fly at a constant speed and altitude. But if the child begins to lose focus, the airplane may slow down. By giving feedback in real time, the child's brain can begin to recognize these patterns and regain focus. Over a few dozen sessions, the child will be able to focus for longer periods of time in everyday life—something that medication alone cannot do [8].

Neurofeedback relies on the principle that brain function can be self-regulated, which will be explored further in the following section.

1.2 Neural Plasticity

It has been shown in several animal studies that the brain can control neuronal activity through operant conditioning. As a type of associative learning, operant conditioning is a process in which the operant's behavior is modified and associated with a stimulus [9]. One theory that undergirds this phenomenon is known as Hebbian plasticity, which is widely agreed as the mechanism through which new information is learned

and retained in the brain [10]. Hebb proposed that when a pre-synaptic neuron frequently fires at a post-synaptic neuron, the synapse itself is altered so that the post-synaptic neuron has a higher probability of undergoing excitation [9].

Operant conditioning has been observed extensively in animals. In one study conducted on rhesus monkeys, researchers found that the monkeys could control the activity in the frontal eye field (FEF), located in the frontal cortex of the brain, in response to juice rewards for adjusting the firing rates of those neurons [11]. Another study done in rats required the animals to alter activity in the primary motor cortex (M1) in order to receive a sucrose reward [12]. These studies show that neurons can adapt their electrical activity in response to external stimuli.

Several studies done in humans have shown similar results [9]. The principle of neuroplasticity has been applied to several neurofeedback experiments done in humans that demonstrate changes in cortical excitability. In one study, transcranial magnetic stimulation (TMS) was applied to the motor cortex of the brain. The researchers found sustained changes in excitability of the motor cortex, lasting at least 20 minutes [13].

In animals and humans alike, regions of the brain can be altered in their excitability and synaptic connections in response to stimuli. Neurofeedback takes advantage of this fact by measuring EEG signals and converting them to a sensory stimulus. Where does EEG come from, and how is it measured? This will be discussed at length in the following section.

1.3 Background on EEG

Like other organs in the body, the brain produces a bioelectric signal that can be recorded as an EEG. The EEG records currents in the brain produced by synaptic excitation as a result of neuronal action potentials [14]. These currents are produced by the pumping of sodium, potassium, calcium, and chloride ions across cell membranes as neurons experience rapid depolarization. Electroencephalography detects the corti-

cal activity underneath the skull on the surface of the brain. The aggregate of these currents can be measured at various points on the forehead and scalp using metal electrodes, and the resulting signal is amplified and digitized for analysis. Figure 1 shows the international 10–20 system of EEG electrode placement at various points on the forehead and scalp; this allows for consistent placement of the electrodes.

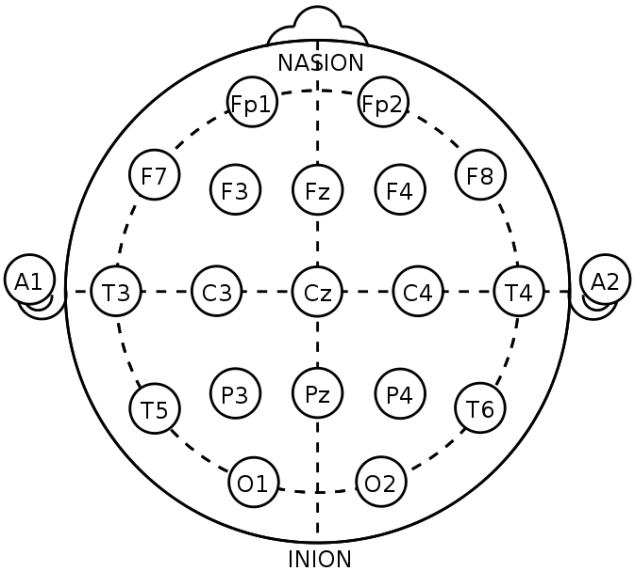


Fig. 1. 10–20 system of EEG electrode placement [15].

The EEG signal is on the order of 0.5-100 μV and generally has a sinusoidal shape [14]. It is a composite of many waves of different frequencies. Thus, the types of waves can be categorized by frequency range: delta (0.5-4 Hz), theta (4-8 Hz), alpha (8-13 Hz), and beta (13-30 Hz). Gamma waves contain frequencies larger than 30 Hz. As shown in Figure 2, the frequency and amplitude varies with each frequency band.

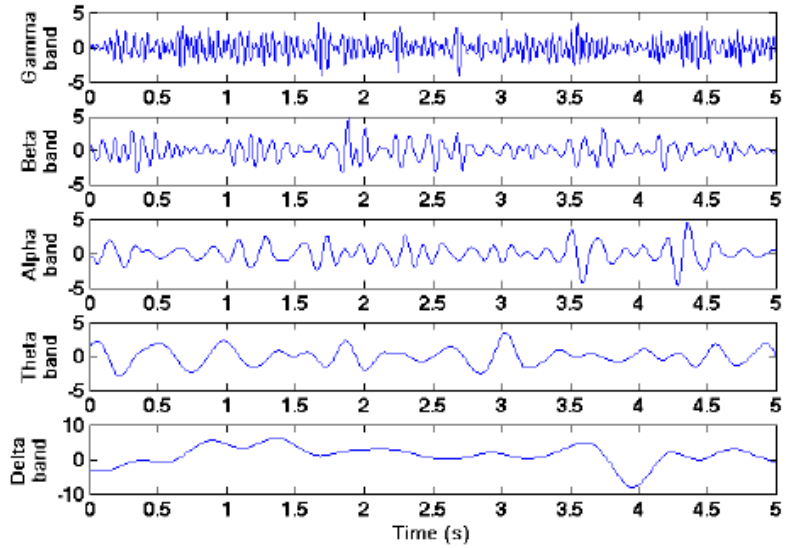


Fig. 2. Classification of EEG waves [16].

Higher-frequency beta waves are the predominant brainwave in awake, alert people and can be measured throughout the frontal cortex. Beta waves range in frequency from 13 Hz to 30 Hz, and can be further divided into three smaller ranges, the sensorimotor rhythm (13-15 Hz), beta-1 (15-20 Hz), and beta-2 (20-30 Hz). Sensorimotor rhythm, or SMR, is associated with mental alertness. Beta-1 corresponds to thinking and attention, and beta-2 is associated with hyperactivity and anxiety [18].

Alpha waves originate in the occipital lobe and are found both in wakeful and sleeping states. In periods of consciousness, alpha waves increase in power during wakeful relaxation with closed eyes [19]. Many studies have shown that alpha power increases during creative ideation processes [20]. Alpha waves also appear during rapid eye movement (REM) sleep, although the exact role of these waves in sleep is not yet known [19]. Alpha waves appear around the age of three in normal subjects [21].

Theta waves correspond to a range of psychological states, though their complete function remains enigmatic. They are associated with drowsiness, deep relaxation, depression, and creativity [18]. In humans, two different types of theta waves are observed: hippocampal and cortical theta rhythms, which are thought to have independ-

ent mechanisms [19]. Theta power changes as humans age, and they are believed to play a large role in thinking, memory recall, and learning [22]. One study found that damage to the hippocampal cortex, which disturbs theta rhythms, resulted in loss of spatial memory [23]. However, more research must be done to fully appreciate the function of theta waves.

Lastly, delta waves are associated mostly with sleep. They originate in the thalamus and are the dominant frequency during deep, non-REM sleep [24]. Research has shown that delta power is highest in infants, who spend a large amount of time sleeping; delta power subsequently decreases throughout life due to aging [25]. Certain parasomnias, such as sleepwalking and sleep terrors, have been linked to disturbances in delta waves [26].

By determining the frequency components of an EEG signal, the type of brainwaves present can be deduced. This is most commonly done with a Fourier transform, which transforms a signal in time to the frequency domain. However, Fourier transform assumes a stationary signal; in the case of EEG and other bioelectric recordings, the signals are nonstationary. For greater precision, a nonlinear form of analysis such as Higuchi's fractal dimension method may be used, although this is much less common in the literature [17]. Once the EEG has undergone some type of frequency analysis, the corresponding mental state of the patient can be predicted.

Thus, analyzing an EEG signal is critical in conducting neurofeedback treatment. The neurofeedback system must be able to calculate the power of each frequency range in a patient's EEG, since each range corresponds to particular mental states. The goal of neurofeedback is to increase the power of a specific range, which induces the desired mental state. In the treatment of ADHD, it has been shown that SMR waves (13-15 Hz) are associated with increased focus, while theta waves (4-8 Hz) correspond to a state of hyperactivity and inattentiveness [27]. Thus, the goal of administering neuro-

feedback to treat ADHD would be increasing SMR power while decreasing theta power. Neurofeedback has been found highly efficacious and specific for the treatment of ADHD and is rated as "Level 5" out of five possible levels [27].

Other brain-related conditions can also be treated using neurofeedback. In clinical depression, patients experience hypometabolism in the cingulate cortex, frontal cortex, and basal ganglia [18]. Depressed patients also exhibit alpha wave asymmetry in the frontal cortex, in which alpha power is higher on the left side [28]. In the treatment of depression, the goal of neurofeedback is to increase the signal power of both alpha waves (8-13 Hz) and theta waves (4-8 Hz) [29].

Children with autistic spectrum disorder (ASD) may also benefit from neurofeedback treatment. The most common form of ASD is characterized by high beta activity. Neurofeedback for ASD often involves enhancing beta while decreasing alpha or theta. In one study, enhancement of 13-15 Hz waves and reducing 3-10 Hz waves resulted in improvement in academic and social functioning [30].

1.4 Problem Statement

Although neurofeedback is a revolutionary therapy that has the potential to treat several brain-related conditions, it is expensive and requires trips to a doctor's office. An average course of treatment, which includes around 30 sessions, costs about $3,000 and is not covered by many insurance plans [31]. The patient must visit a doctor's office three to four times a week for 30-45 minute sessions; for those suffering from certain affective disorders, such as depression, leaving the home can be an insurmountable burden.

Before forming a solution to this problem, we looked at existing products on the market that use EEG signals to see if any devices currently address this need.

Existing EEG products. One such device is the Muse Headband™, which provides binaural beats to assist with meditation (shown in Figure 3). It includes five dry electrodes placed between Fp1, FpZ, and Fp2, with the ground electrode placed in the middle of the forehead. However, it does not use medical-grade electrodes and is designed for recreational use. It costs $250.

Fig. 3. The Muse headband [32].

Another device is the NeuroPlus Headband™, which uses the EEG signal in conjunction with a tablet-based app to improve focus in children (shown in Figure 4). It uses a single claw dry electrode with eight legs, which is positioned at the top of the head (Cz) [33]. Although this concept is similar to neurofeedback, the device is not intended for therapeutic use; the website specifically states that the device cannot be used in the treatment of ADHD. Furthermore, the device uses dry electrodes, which are not recommended for medical use. It costs $99 for the headband and a monthly fee of $30 for access to the mobile app that is used to play the games.

Fig. 4. The NeuroPlus headband [34].

Finally, the EMOTIV Insight is a headband device used for self-quantification of brain activity (shown in Figure 5). The software allows users to observe brain visualization images to improve cognitive function. It uses five dry electrodes placed at AF3, AF4, T7, T8, Pz, with the ground on the left mastoid process [35]. The five-channel device costs $299, while the 14-channel device costs $799. The accompanying mobile app is free, although users can also buy more advanced software that costs up to $99 per month.

Fig. 5. The EMOTIV Insight [35].

1.5 Objective

Based on the research and available products, we identified a need for a lower-cost form of neurofeedback. Our objective is to create an affordable device that will allow

users to conduct clinical-grade neurofeedback treatment from the comfort of their own homes.

2 System Overview

2.1 System Requirements

To address the diverse needs of all the people who could benefit from neurofeedback treatment, our system will be customizable for each user. The device will be a comfortable, wearable product that allows for feedback in real time. It will transform a bioelectric signal into a sensory signal (audio, visual, tactile, or some combination).

2.2 Specifications and Constraints

Based on our objectives, we developed a set of specifications corresponding to desired system requirements. These are summarized in Table 1.

Table 1. Specifications Summary

Requirement	Specifications	Reasoning
Affordable	Price ≤ $250	< cost of 3 sessions
Wearable	Weight ≤ 550g, Temperature ≤ 80˚F, Low power	Comfort, sustained use
Fast	Responds to a change in brain frequency in ≤ 1s	Real-time feedback
Accurate	Able to distinguish brainwave frequencies within 1 Hz	Provide correct feedback

First, we intended for the device to be affordable relative to traditional neurofeedback therapy. We set a price limit for production at $250, which corresponds to the typical cost of three therapy sessions. Next, the device was to be wearable. In order to achieve user comfort, the device had to be lightweight and stay cool to the touch. As a

wearable, the device should also be low power, such that use could be sustained without frequent replacement of the system's batteries. Another requirement was system speed. To accomplish real-time feedback, we determined that the device should respond to changes in brain frequency within one second. Finally, we wanted the system to be accurate. We decided that in order to provide correct feedback, the system should be able to distinguish brain frequencies within 1 Hz of each other.

2.3 Block Diagrams

To create such a device, we divided the process of capturing a signal and providing real-time feedback into four distinct phases. Figure 6 shows each system block and its components.

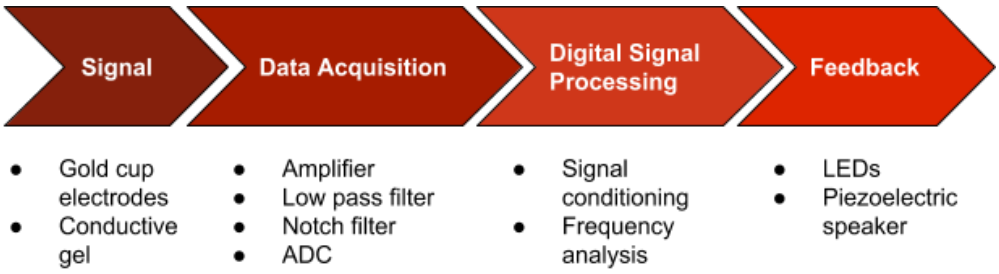


Fig. 6. System block diagram.

In the first stage, the EEG signal must be collected using metal electrodes and a conductive medium. Next, the signal must undergo amplification, filtering, and conversion from analog to digital. This renders the signal viewable on a digital interface. The digital signal is then conditioned and analyzed to determine its frequency components. A simplified audiovisual feedback in the form of colored light emitting diodes (LEDs) and a piezoelectric speaker is then provided based on the analyzed data. The following section will discuss each system block in detail.

3 Design

3.1 Chosen Implementation

Electrodes. The first stage of the device involves collecting the EEG signal, which is done using electrodes and a conductive medium. The choice of electrode is critical for recording EEG potentials, which are extremely low-amplitude and can be noisy. A study found that the best type of electrode for recording EEG is sintered Ag/AgCl electrodes, which are made using a process that allows better adhesion between the silver and silver chloride [36]. However, these electrodes are expensive and cost $40 each, on average. We decided to use OpenBCI gold cup electrodes, which cost $10 a pair; these were rated as "Good" on a scale of "Poor" to "Excellent" for AC-coupled long-time recording of EEG [36]. Gold cup electrodes are similar to the kinds used in many medical devices, as they ensure better contact with the skin and better conductivity than dry electrodes. Dry electrodes have been observed to have higher contact impedance and lower signal quality than gel electrodes [37].

Metal electrodes need to be used with a conductive electrolyte gel. Research on different kinds of EEG gels showed that Ten20™ conductive EEG paste is the most effective in gathering a signal [36]. We ordered this gel and used it to test the final prototype.

Amplifier Design. Because the signal gathered by the electrodes is so small (~100 μV), it must be amplified before it can be analyzed on a digital interface. The signal is measured as a differential across two electrodes; thus, a one-channel EEG system contains two electrodes. For this reason, an instrumentation amplifier is desirable to amplify the differential signal. To mitigate the electrically noisy environment, the instrumentation amplifier must have high common mode rejection of at least 100 dB, as well as high input impedance of at least 100kΩ [14].

Common mode rejection is crucial for reliable measurement of any biopotential signal. For an ECG signal, which is generally on the order of 1 mV, an acceptable amount of interference is between 1-10 μV peak-to-peak [38]. One way of dramatically reducing the common mode voltage is a right-leg driver circuit. This circuit uses a third electrode that attaches to the human body, which "provides a low-impedance path between the patient and the amplifier common" [39]. The circuit contains a voltage buffer that "drives" the user to have the same voltage as the amplifier common, thereby increasing the CMRR. Thus, it was desirable to select an instrumentation amplifier that could easily be connected to a right-leg driver circuit. Figure 7 shows the implementation of a right-leg driver circuit in an ECG application.

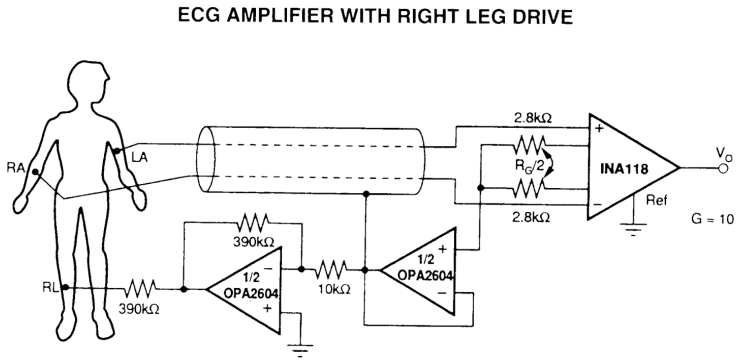


Fig. 7. Schematic of a right-leg driver circuit in ECG application [40].

Based on these considerations, we chose the Texas Instruments INA-826 instrumentation amplifier board [41]. The schematic side of the circuit board is shown in Figure 8, and the specifications are shown in Table 2.

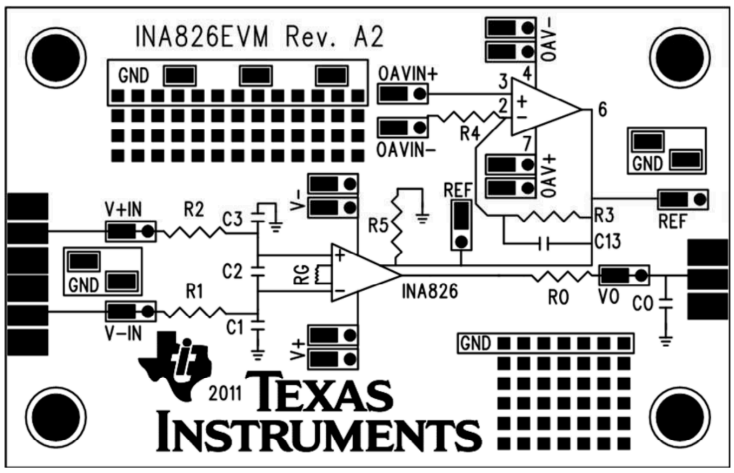


Fig. 8. INA-826 evaluation board, schematic side [41].

Table 2. Electrical specifications of INA-826 instrumentation amplifier [41].

Specification	Value
Gain	100 V/V
CMRR	100 dB
Common-mode Impedance	20 GΩ

It has a programmable gain, high common mode rejection of 100 dB, and a right-leg driver circuit. The gain was set to 100 with a gain resistor. A low pass-filter can be implemented at the input by choosing values for R1, R2, C1, C2, and C3.

To help decrease the effect of noise from the environment, especially 60 Hz interference, we implemented the low-pass filter at the input of the circuit. Equation 3.1, taken from the INA-826 EVM datasheet, can be used to design the low-pass filter. It is assumed that R1 = R2, C1 = C3, and C2 is about ten times larger than C1 and C3.

$$f_c = \frac{1}{2\pi(R1+R2)(C2+\frac{C1}{2})} \qquad \textbf{(3.1)}$$

The cutoff frequency was set to 40 Hz, because EEG signals fall within the range of 0.5 Hz to 30 Hz. With a cutoff of 40 Hz, we could ensure minimal attenuation for the beta frequencies. For R1 and R2, a value of 8.66 kΩ was chosen. Thus, C1 = C2 = 0.22 nF, and C3 = 2.2 nF.

The right-leg driver circuit was implemented according to the specifications in the INA-826 EVM datasheet [41]. Figure 9 shows the design of a right-leg driver circuit with a single-supply configuration on the evaluation board. The values of C4, C5, C11, and C12 in Figure 9 were the same ones used in the prototype.

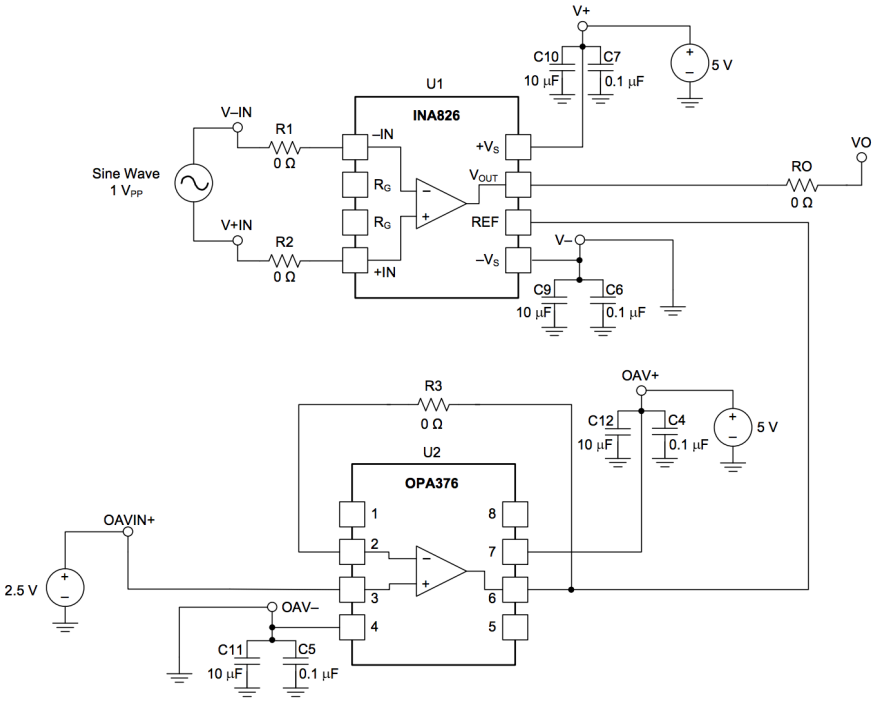


Fig. 9. Schematic of right-leg driver circuit on INA-826 evaluation board [41].

The operational amplifier selected for the right-leg driver circuit was the Analog Devices OP-281. It is a dual op-amp with low power and low noise. Table 3 lists some of the specifications for the op-amp.

Table 3. Electrical specifications of OP-281 op amp [42].

Specification	Value
Gain	1
Noise Density	75 nV / √Hz
Supply Current	4 μA
CMRR	90 dB

At the output of this circuit, the EEG signal has been amplified by a factor of 100 and undergone preliminary filtering. The goal of the following stage is to further amplify and filter the signal.

Low-Pass Filter Design. Recording biopotential signals results in power-line interference of 50 Hz (Europe and Asia) or 60 Hz (United States). The capacitance between the body and the earth couples with the capacitance between the body and the power main, causing an interference current that flows from the power line through the body to the earth. This current, which is generally about 0.5 μA peak-to-peak, can partially flow back to ground through the right-leg driver circuit [38]. However, this is often not enough to fully remove power line interference. Thus, a low-pass filter can be implemented to further filter the noise. Additionally, the EEG signal must be amplified by at least 10,000 from its original amplitude of ~100μV. Thus, another goal of this stage is to amplify the signal by another factor of 100.

A second order Sallen Key low-pass filter was designed. The Sallen-Key topology is advantageous for its simplicity and cost-effectiveness, as it requires only one operational amplifier. The main drawback of this topology is its relatively limited quality factor compared to other designs, such as a biquad filter; this makes the roll-off less sharp [43]. However, because the EEG signal will be further filtered by both analog and digital means, a simpler design was more appropriate for our application.

The filter was implemented on an Analog Devices Sallen-Key low pass filter evaluation board, EVAL-FW-LPSK2. The evaluation board is a pre-designed PCB that allows soldering of resistors, capacitors, and an operational amplifier. Two resistors at the op-amp output set the gain to 100. The design was created using Analog Device's online Filter Wizard. Figure 10 shows the schematic for the low-pass filter.

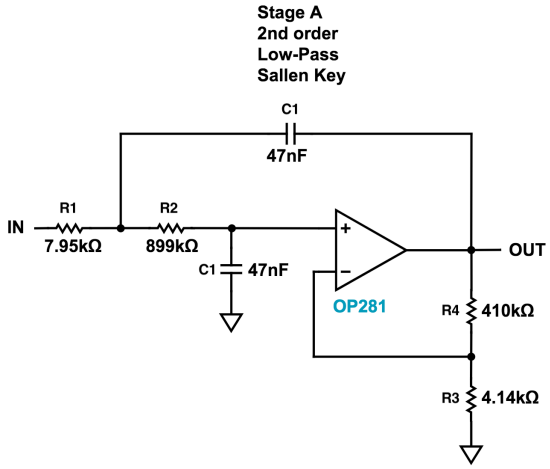


Fig. 10. Schematic of second-order low pass filter.

The Filter Wizard also generated a Bode plot for the filter, which is shown in Figure 11. The cutoff frequency was 40 Hz, and the -10 dB frequency was 70 Hz.

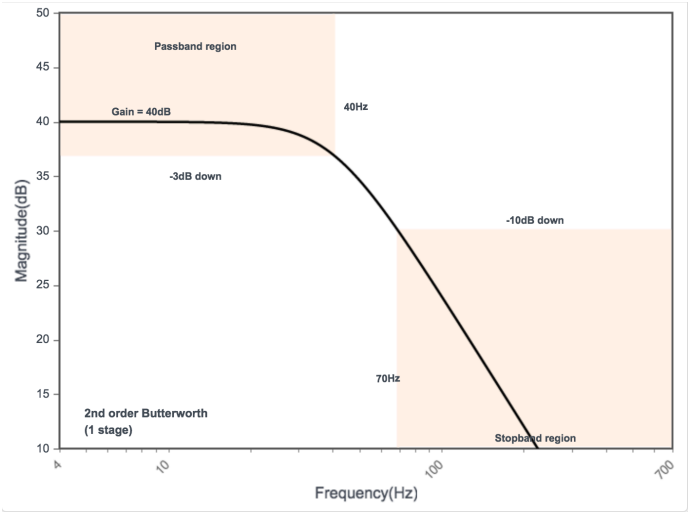


Fig. 11. Bode plot for second-order filter.

The Analog Devices OP-281 operational amplifier, which was the same one used for the right-leg driver circuit, was used for the filter circuit as well. This was recommended by Analog Devices's Filter Wizard; furthermore, the low-power and low-noise characteristics made it an advantageous choice for this application.

The actual values of the components used are shown in Table 4.

Table 4. Component values for low pass filter [42].

Component	Design Value	Actual Value
R1	7.95 kΩ	7.87 kΩ
R2	899 kΩ	909 kΩ
R3	4.14 kΩ	4.12 kΩ
R4	410 kΩ	412 kΩ
C1	47 nF	47 nF
C2	47 nF	47 nF

At the output of this circuit, the EEG signal has been amplified by a factor of 10,000 and filtered through a low-pass filter. Next, the signal will undergo further filtering.

Notch Filter Design. Even with the instrumentation amplifier filter and Sallen-Key filter, the EEG signal still contained a high amount of power line interference that obscured the signal itself. One solution would be a higher-order low-pass filter, but these take up a lot of area and are not as appropriate for a wearable device. Instead, most EEG applications contain a notch filter for further elimination of power line interference [44]. A notch filter allows low frequencies to pass while dramatically attenuating the 60 Hz noise.

The most common notch filter topology is the Twin T notch filter, which consists of two resistors and two capacitors. Essentially, it is a combination of a low-pass and high-pass filter. The schematic for this circuit is shown in Figure 12.

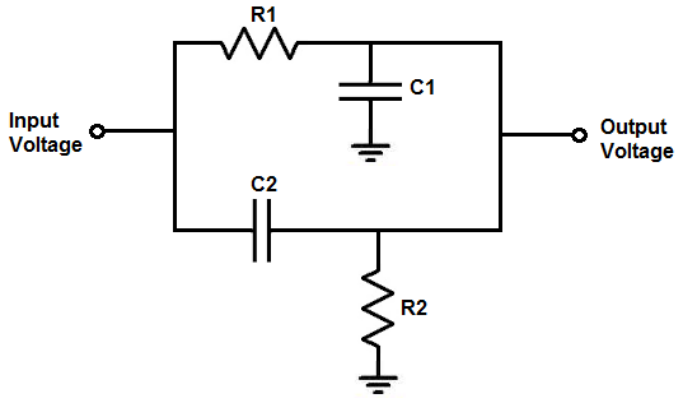


Fig. 12. Schematic of passive Twin T notch filter [45].

However, a large drawback of the passive Twin T filter is the limited quality factor. The Q-value, or quality factor, is fixed at 0.25, making the roll-off less sharp at the center notch frequency. This limitation can be overcome using an active Twin T notch filter, which includes positive feedback at the reference. The schematic for this filter is shown in Figure 13. The Q-value, which corresponds to the depth of the notch filter, can be tuned with the ratio of R4/R5. Alternatively, R4 and R5 can be replaced with a potentiometer, which allows for manual tuning of the Q-value.

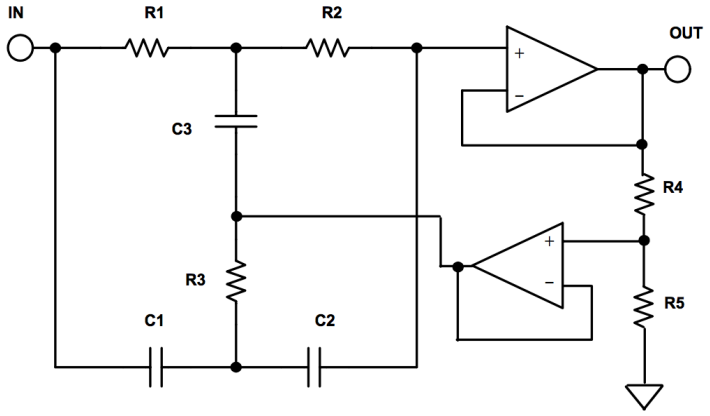


Fig. 13. Active Twin T notch filter [46].

Because a large attenuation of the 60 Hz interference was desired, the active Twin T configuration was selected. For our application, the notch filter was designed on a

SparkFun Op-Amp breakout board, LMV358. To calculate the center notch frequency of a notch filter, Equation 3.2 can be used [47].

$$f_c = \frac{1}{2\pi RC} \tag{3.2}$$

For a notch frequency of 60 Hz, a capacitor value of 7.4 nF was selected. Using Equation 3.3, the calculated value of R was 360 kΩ.

$$\frac{1}{2\pi * 360e3 * 7.4e-9} = 54.74Hz \tag{3.3}$$

With respect to Figure 13, R=R1=R2, R3 = R/2, C=C1=C2, and C3=2C. Component matching is important to achieve a deeper notch [48]. Figure 14 shows the design of our notch filter. Like in the second-order low pass filter and the right-leg driver circuit, the Analog Devices OP-281 operational amplifier was used. Because it is a dual op amp, only one IC was required. Instead of using R4 and R5, a 1M potentiometer was used to vary the depth of the notch filter.

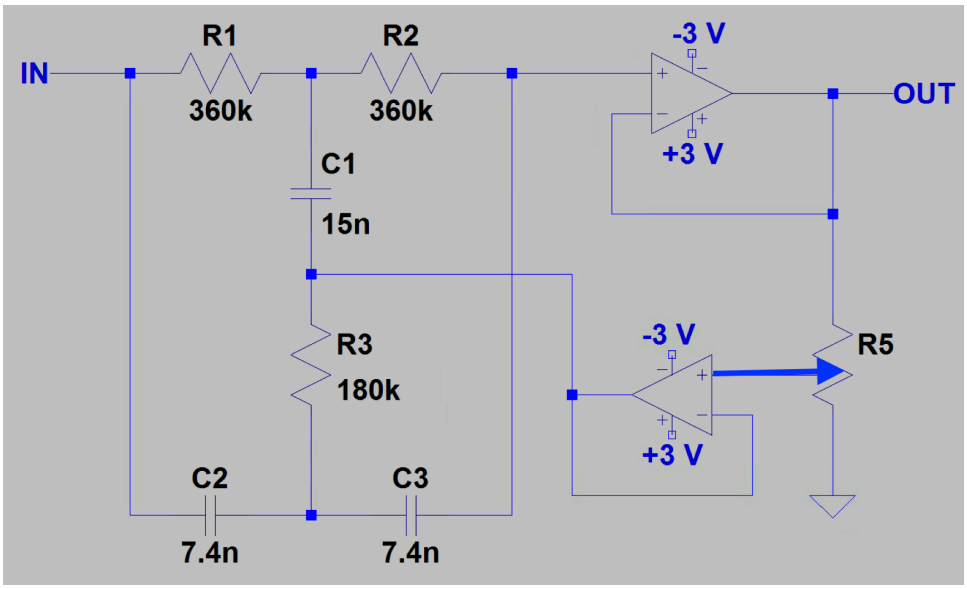


Fig. 14. LTSpice model of notch filter circuit.

At this output of the notch filter, the EEG signal has been sufficiently amplified and filtered, making it ready for digitization.

ADC and Digital Signal Conditioning. After completing analog processing, the input signal was converted to digital using an Arduino Mega2560's onboard Analog-to-Digital Converter (ADC). This ADC has a 10-bit resolution and operates at 9.6kHz, much faster than needed to sample brain waves, which fall between 0.5 and 30 Hz. In order to maximize digital signal resolution, we set an external reference voltage of 0.825V. This spread the ADC's 1,024 levels over a smaller voltage range, better capturing the full signal with a resolution of 0.81mV/level (Eq. 3.4). After plotting an incoming brainwave, we noticed a slight DC bias in the signal. In order to correct for this, we digitally shifted the signal down by 0.1V to re-center the signal around zero.

$$ADC\,\text{Re}solution = \frac{V_{ref}}{2^{(N\,bits)}} \qquad \textbf{(3.4)}$$

Frequency Analysis. The goal of this subsystem was to isolate the dominant frequency of an incoming brainwave in real time. To accomplish this, we performed a Fast Fourier Transform (FFT) on the incoming signal. The FFT is an algorithm that samples a signal over a period of time and divides it into its frequency components.

In order to implement the FFT, we used an open-source library, arduinoFFT [49]. We selected a sampling frequency of 200Hz; this is well above the 60 Hz Nyquist rate, which is two times the highest frequency in the EEG signal. In order to meet our frequency resolution specification, we took a 256 sample FFT. This gave us a resolution of 0.78Hz, but increased algorithm run-time to 1.28 seconds, as calculated in Eq. 3.5 and 3.6. Increasing the number of samples to the next power of two, 512 samples, would have further increased frequency resolution at the expense of speed; however, this higher sample size overflowed the Arduino's memory when tested.

$$Runtime\ speed = N\ samples * \frac{1}{f_s} \quad \textbf{(3.5)}$$

$$Frequency\ Bin\ Resolution = \frac{f_s}{N\ samples} \quad \textbf{(3.6)}$$

Feedback. After performing signal frequency analysis, we further processed the FFT output to obtain information about the four brainwave frequency ranges: delta, theta, alpha, and beta. We also examined signal noise, which we considered to be everything above 30 Hz—the maximum frequency of beta waves. This processing involved using the magnitude of each frequency bin, as calculated by the FFT, to find the signal power of each of the brainwave frequency ranges.

$$Bin\ Frequency = \frac{i \cdot f_s}{N\ samples} \quad \textbf{(3.7)}$$

$$Frequency\ Range\ Power = \sum_{n}^{k} \left| X_n \right|^2 \quad \textbf{(3.8)}$$

Frequency range signal power was calculated using Eq. 3.8, where *n* and *k* are bin frequencies corresponding to starting and ending frequencies of each brainwave range, respectively. Bin frequencies can be found using Eq. 3.7, where *i* ranges from 0 to N/2 samples and f_s is the sampling frequency. After calculating the power of each frequency range, the relative and percent signal powers were computed as shown in Eq. 3.9 and 3.10 to determine incoming brainwave frequency content.

$$Relative\ Signal\ Power = 10\log_{10}\left(\frac{Frequency\ Range\ Power}{Total\ Signal\ Power}\right) \quad \textbf{(3.9)}$$

$$Percent\ Signal\ Power = \frac{Frequency\ Range\ Power}{Total\ Signal\ Power} \cdot 100 \quad \textbf{(3.10)}$$

With these computations, it was possible to determine the dominant frequency of a brain signal. This information was used to light colored LEDs and toggle a piezoelectric buzzer.

3.2 Alternative Implementations

Digital Notch Filter. Instead of implementing an analog notch filter, it would have been possible to use a digital notch filter. However, it has been shown that using a digital notch filter instead of an analog one reduces the resolution of the digital signal [44]. Thus, an analog notch filter was more suitable.

Microcontroller. An Arduino microcontroller was selected to perform analog-to-digital conversion, frequency analysis, and audiovisual feedback functionality. It was chosen due to extensive documentation and support for Arduino projects as well as our previous experience using an Arduino board and its IDE. After narrowing board options down to the Uno and the Mega2560, we opted for the Mega2560 due to its greater memory.

ADC. Upon initial research of ADCs for bioelectric signal collection, we considered the Texas Instruments ADS1291, a low-power, 24-bit chip for ECG (electrocardiogram) applications. However, we determined it was cheaper and possible to get a similar resolution using a 12-bit ADC with an external reference voltage to better use the full range of the ADC. We selected the MAX11102, with a minimum V_{ref} of 1V. After interfacing it with the Arduino Mega2560, we decided to simplify the system by eliminating use of an external ADC and using the Mega2560's onboard 10-bit ADC. In order to maintain high signal resolution, we used a voltage divider on the Arduino's regulated 3.3V output to set V_{ref} to 0.825V.

4 Testing and Results

4.1 Oscillator Test Circuit

Preliminary testing of the circuits was done using a digital function generator as the input. However, the lowest amplitude signal that can be produced is a 10 mV peak-to-peak sine wave. While this input signal would provide some insights into the functionality of the circuit, it is not a good model of a real EEG signal. A human brain signal has a peak-to-peak voltage of about 100 μV. Thus, a simple oscillator circuit was designed to provide a brain-like signal that could serve as the input to the instrumentation amplifier.

A square-wave oscillator, also known as an astable multivibrator, generates a periodic square-wave signal. It has a fairly simple topology, and the frequency of the output signal can be easily tuned with the choice of resistor. Figure 15 shows the schematic for a square-wave oscillator.

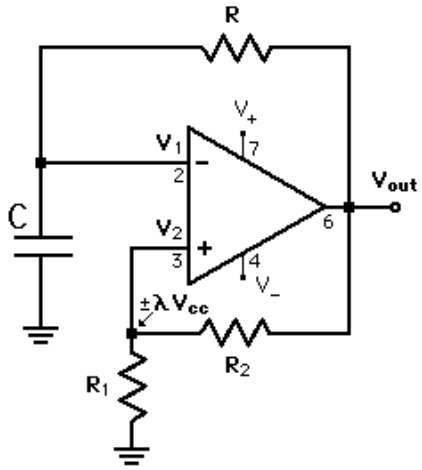


Fig. 15. Schematic of a square wave oscillator [50].

Because we needed to analyze the behavior at the circuits for various input frequencies, the oscillator needed to have a tunable frequency. Furthermore, the output voltage needed to be small enough to mimic a brain signal. Thus, two variable resistors were used: one to vary the frequency of the output (R1) and one to vary the amplitude

of the output (R5). The potentiometer values were chosen so that the frequency would vary between 1 Hz and 80 Hz, and the amplitude could vary between 50 µV and 3 mV. Figure 16 shows an LTspice model for the oscillator.

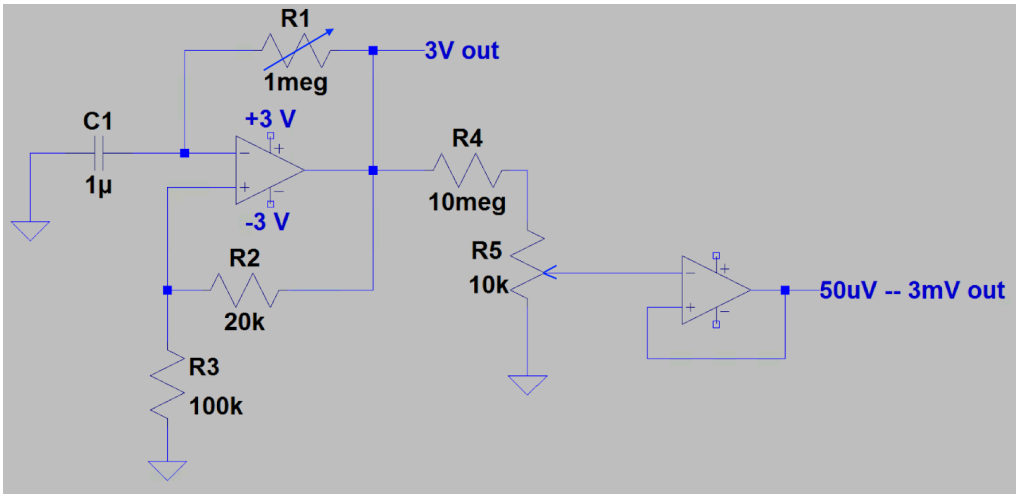


Fig. 16. LTSpice model of oscillator circuit.

The oscillator circuit was used in the hardware testing, described in the following section.

4.2 Hardware Testing

Amplifier Testing. Hardware testing began with the INA-826 EVM to assess the functionality of the low-pass filter and to verify the gain. One of the differential inputs was connected to ground, while the other was connected to the output of a digital function generator. The output from the generator was a sinusoidal wave with an amplitude of 10 mV and a frequency that varied from 1 Hz to 100 Hz. The voltage at the output was measured with each change in frequency. Figure 17 shows how the magnitude of the output voltage varied with frequency due to the low-pass filter.

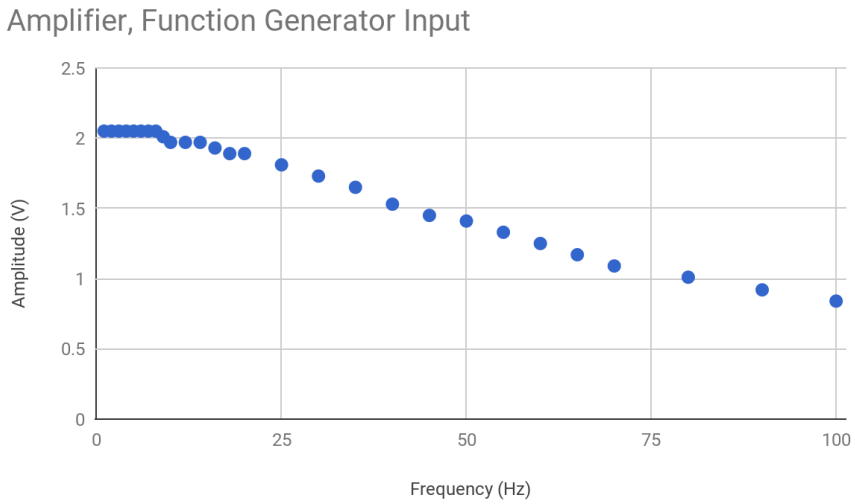


Fig. 17. Amplitude of output voltage vs input frequency for a function generator input.

As Figure 17 shows, the signal was amplified to approximately 2 V at low frequencies, between 1 and 8 Hz. As the frequency of the input signal increases, the magnitude of the output voltage decreases. At 40 Hz, the signal attenuates to 1.53 V. At 60 Hz, the output was 1.25 V. This showed that the low-pass filter successfully attenuated frequencies higher than 40 Hz, although the attenuation was not very large.

The amplifier was then tested using the oscillator circuit. The input was set to 3 mV, and the frequency was varied from 1 Hz to 80 Hz. Figure 18 shows the change in amplitude of the output signal as the input frequency increases.

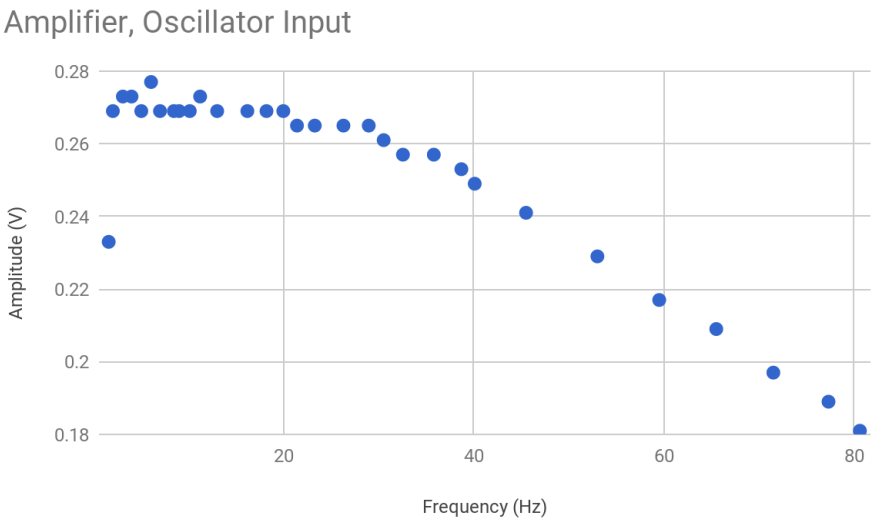


Fig. 18. Amplitude of output voltage vs input frequency for oscillator input.

The oscillator signal was amplified to about 0.27 V at low frequencies, which is almost exactly 100 times greater than the original 0.003 V signal. At 40 Hz, the signal had attenuated to 0.25 V, and at 60 Hz, the signal had an amplitude of 0.22 V.

Low-Pass Filter Testing. The next phase of testing involved the second-order low-pass filter board. Like the instrumentation amplifier, testing was first conducted using the digital function generator. The input was a 10 mV peak-to-peak sine wave with frequency that varied between 1 Hz and 100 Hz. The amplitude of the output was measured in volts and graphed with respect to the input frequency, shown in Figure 19.

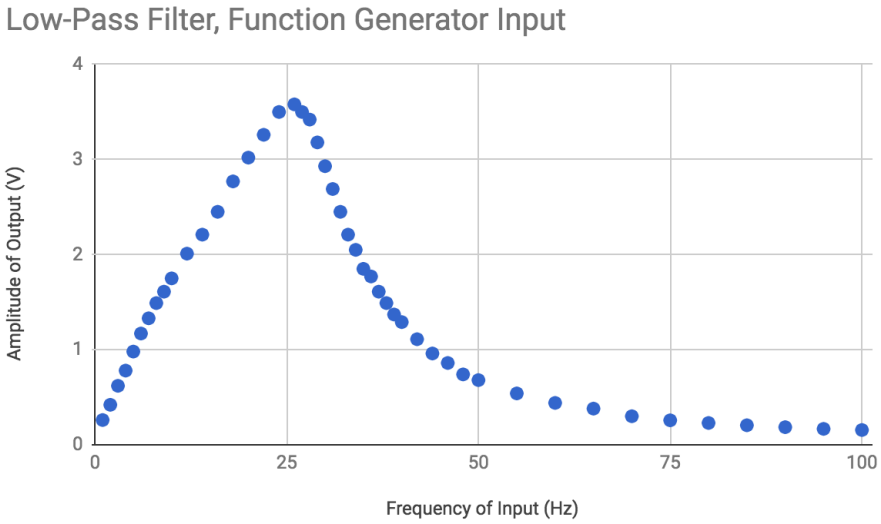


Fig. 19. Amplitude of output voltage vs input frequency for function generator input.

Although the filter did attenuate frequencies above 40 Hz, it seemed to also attenuate signals with lower frequencies. The output reached peak amplitude of 3.58 V at 26 Hz; the amplitude at the cutoff frequency (40 Hz) was 1.29 V. At 60 Hz, the signal had attenuated to 0.44 V. The filter amplified the 10 mV signal with a gain of well over 100, which should have given an output voltage of 1 V.

The circuit was then tested using the analog oscillator circuit as the input. The signal amplitude was 3 mV, which was verified through measurement using an oscilloscope. The frequency of the signal varied between 1 Hz and 63 Hz. Figure 20 shows the amplitude of the output voltage as a function of the input frequency.

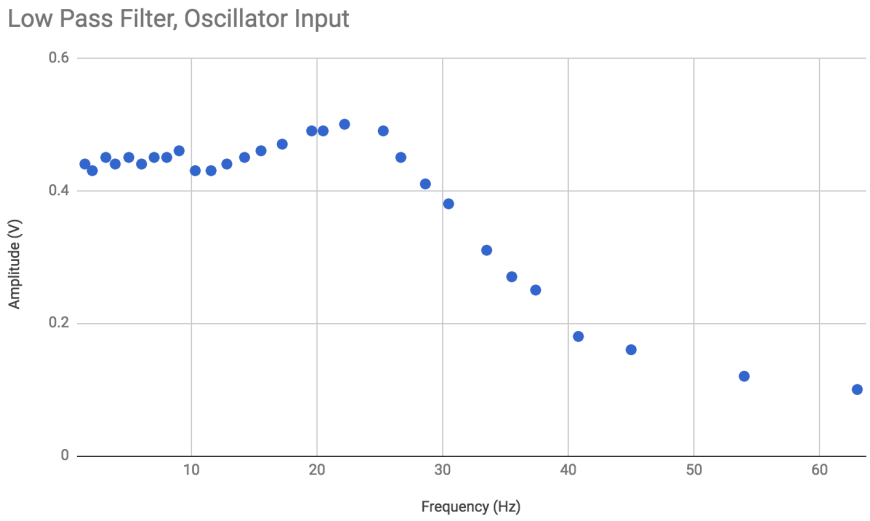


Fig. 20. Amplitude of output voltage vs input frequency for oscillator input.

In this test, the filter operated more closely to its expected behavior. The circuit amplified the signal with a gain greater than 100, since an approximately 0.4 V output signal was obtained for a 3 mV input signal. The signal began to attenuate slightly around 26 Hz, where the output had an amplitude of 0.45 V. At the cutoff of 40 Hz, the amplitude of the output was 0.18 V.

Complete Hardware Testing. The final part of hardware testing involved connecting the output of the amplifier board to the input of the filter board. A signal was connected to the input of the amplifier board, and the peak-to-peak voltage at the output of the filter board was measured. Because the combined gain of the two circuits was 10,000, a digital function generator could not be used in this test. The minimum amplitude that the generator can output is 10 mV, which would have saturated the operational amplifiers.

Thus, the oscillator circuit was used as the input. The oscillator was set to its lowest amplitude output, which is around 50 µV; because the voltage was so small, it was not possible to quantitatively measure it. The frequency of the signal varied between 1 Hz

and 60 Hz. Figure 21 shows the amplitude of the voltage at the filter output, as a function of the input frequency.

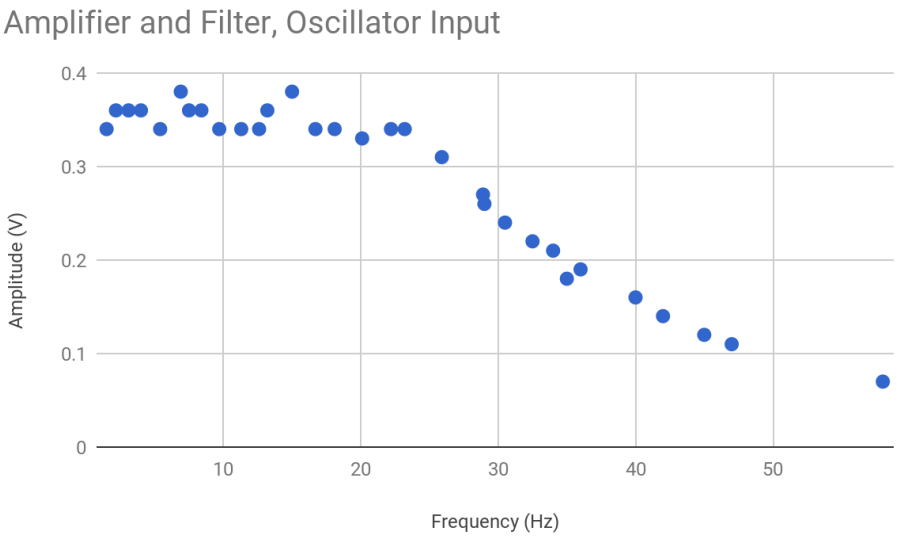


Fig. 21. Amplitude of output voltage vs input frequency for oscillator input.

At low frequencies, the output signal was amplified and not attenuated. With a gain of 10,000, we would expect the 50 µV signal to be amplified to 0.5 V; the output voltage was just under 0.4 V, which verifies the gain of the circuit. The signal begins to decrease in amplitude at 26 Hz. At 40 Hz, the output amplitude was 0.16 V.

4.3 Software Testing

Matlab. Initial software testing consisted of using a function generator in tandem with the Arduino Mega2560's native analogRead() and delay() functions on loop, effectively using delay() to set a sampling frequency with which to sample an input signal. Pure sine and square waves of various frequencies as well as modulated sinusoidal signals were sampled in this manner with the samples being printed to the Arduino Serial monitor in real-time. Data from the Serial monitor was extracted using a program called TeraTerm, which exported Arduino serial data to a .csv (comma separated value) file. The .csv file data was imported into Matlab, and the fft() function

was used to analyze the frequencies of the sampled data. Figure 22 shows Arduino-sampled input data and the corresponding FFT, which matches the expected analysis of the input signal.

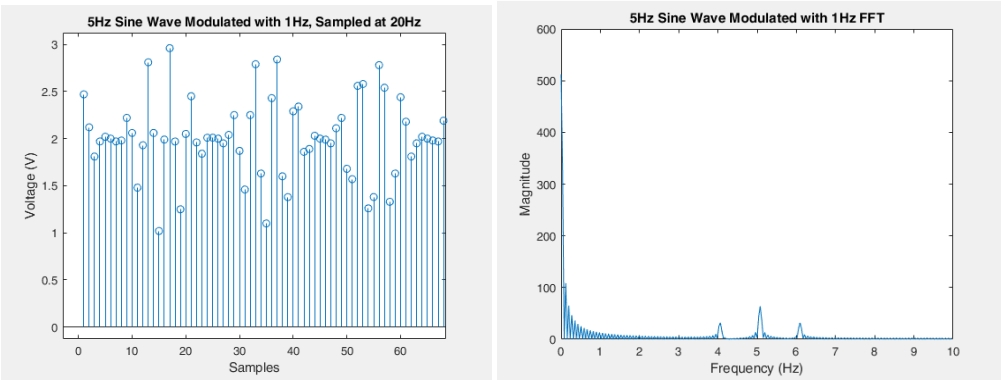


Fig. 22. Sampled Data & FFT Analysis.

While this implementation functioned as expected, it lacked the real-time frequency analysis capability desired for our design. In an attempt to bypass .csv files and external programs to achieve real-time analysis, we installed Matlab's support package for Arduino hardware and connected the Arduino to Matlab directly. When attempts were made to replicate results of the initial software testing using the Matlab support package, delays in signal processing produced inconsistent sampling and FFT results. Because the focus of this project was home usability, we moved away from this implementation in favor of one independent of a Matlab license that could be contained within the Arduino system.

Arduino FFT Library. After looking into various existing libraries for frequency analysis using Arduino, we selected the arduinoFFT library for its straightforward implementation and clear example code. This example code contained multiple display options, including FFT magnitude plot visualization using the Arduino serial plotter (Fig. 23), a serial print of each frequency bin and its magnitude, and a printout of dominant signal frequency as calculated by the library's built-in *MajorPeak* function.

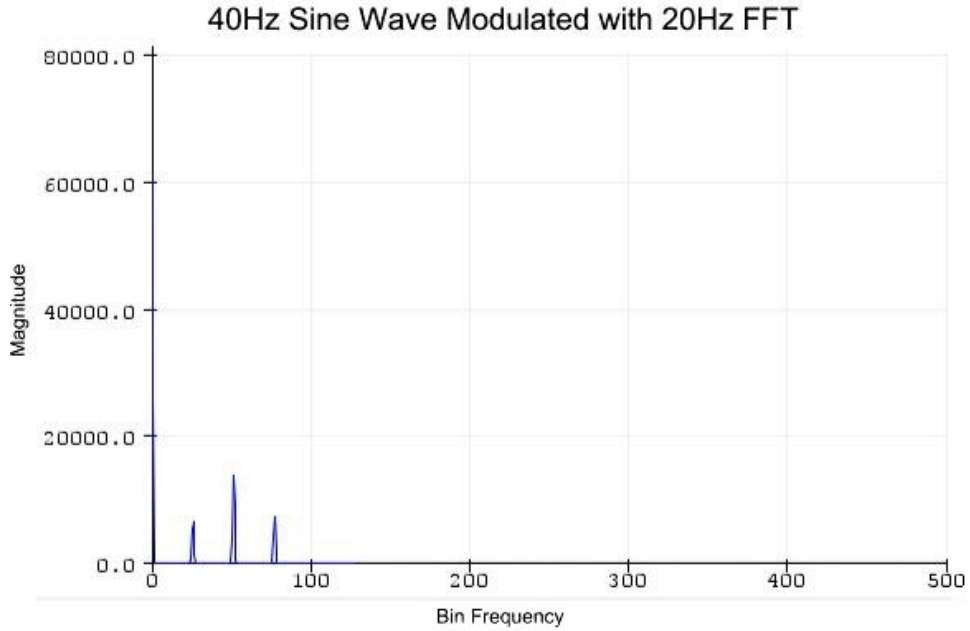


Fig. 23. arduinoFFT Test Result.

After initial testing and library verification with pure and modulated sinusoidal function generator signals, we opted to use the *MajorPeak* function output to determine the highest magnitude frequency in a signal. Using this information, we were able to turn a pair of LEDs on or off if the major peak fell within a specific range. Realizing that this only provided limited information about brainwave readings, we opted to process the output of the arduinoFFT library based on frequency ranges corresponding to delta, theta, alpha, and beta waves. At first, we merely summed the magnitudes of the bin frequencies output by the library in each brainwave range; however, this skewed results in favor of beta waves, as that is the largest frequency range. By squaring the magnitudes of each bin, the effects of bin size are mitigated, as the majority of bins with a magnitude of less than 1 do not significantly contribute to the signal power calculation. With this normalization technique, we were able to take the magnitude of each frequency range and find its power relative to the total signal power, both in terms of dB and percentage of total power. A sample printout of this algorithm can be seen in Fig. 24.

```
Peak = 2.73
RELATIVE SIGNAL POWER:
Delta = -3.46dB
Theta = -6.23dB
Alpha = -7.65dB
Beta = -8.62dB
Other = -28.22dB

PERCENT SIGNAL POWER:
Delta = 45.12%
Theta = 23.83%
Alpha = 17.17%
Beta = 13.73%
Other = 0.15%
```

Fig. 24. Algorithm Relative & Percent Signal Power Sample Output.

Digital Low-Pass Filter. In order to account for any remaining noise in the initial analog signal as well as introduce an anti-aliasing mechanism, we determined it was necessary to implement a digital low pass filter (LPF). We used a 21st order Parks-McClellan filter designed in Matlab for a sampling frequency of 200 Hz. Table 5 contains the filter's band characteristics.

Table 5. Digital LPF Characteristics.

Passband	0-30Hz
Transition Band	30-50Hz
Stopband	50-100Hz

After creating a function to perform a convolution of the incoming signal with Matlab-designed filter coefficients, we tested the filter with function generator signals. We observed the expected behavior of attenuating frequencies above 30 Hz. Next, we used an EEG signal as the input for the function; the pre- and post-filtered signals are shown in Fig. 25.

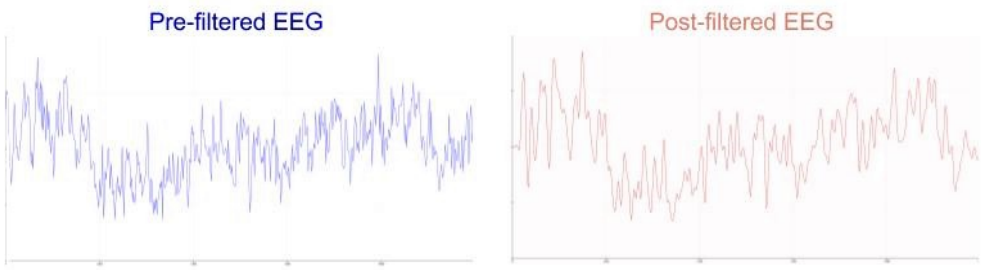


Fig. 25. EEG Signal Pre- & Post-LPF.

Digital Band-Pass Filters. Another method of analyzing the frequency of an EEG signal was to implement digital bandpass filters (BPFs) corresponding to brainwave frequency ranges. These were each 51st order Parks-McClellan finite impulse response (FIR) filters also designed in Matlab. By passing a signal generator input through each filter simultaneously and slowly incrementing signal frequency, we were able to determine if a signal most closely matched delta, theta, alpha, or beta waves based on which filtered signal exhibited the largest amplitude. Test results for both a 6Hz and 10Hz sinusoidal input in Fig. 26 illustrate peaks in theta and alpha waves, respectively.

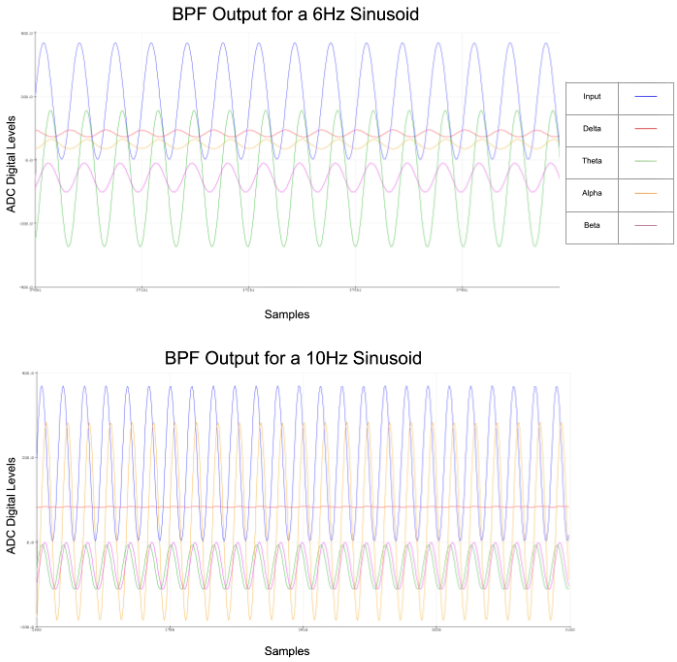


Fig. 26. BPF Test Results.

FFT Algorithm Verification. While the FFT functioned correctly with signal generator signals, we wanted to verify operation when using the algorithm on EEG data. To accomplish this, we found clinical EEG data online and used the samples as the input to the FFT.

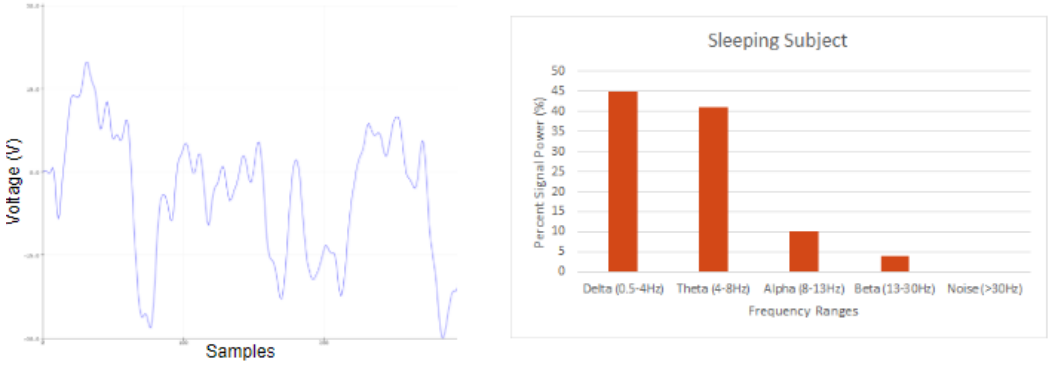


Fig. 27. Sleep Heart Healthy PSG EEG Data & FFT Results.

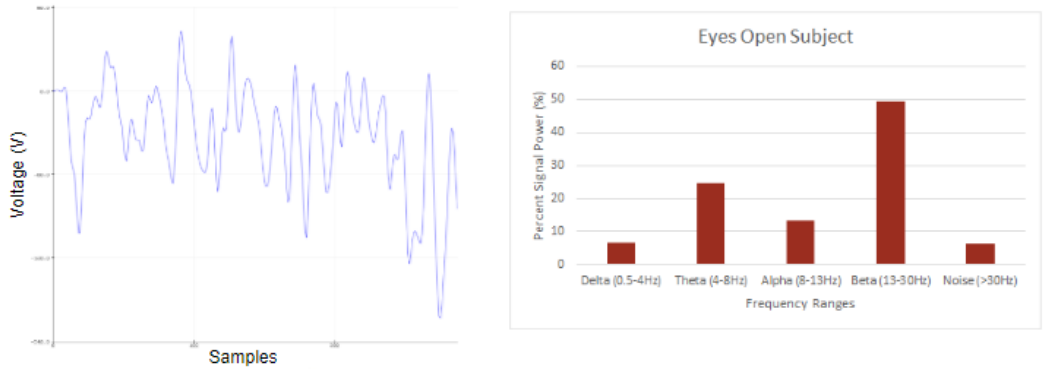


Fig. 28. Healthy Volunteer with Eyes Open EEG Data & FFT Results.

The results in Fig. 27 are from a sleep study, or polysomnography (PSG), and show a peak in delta and theta waves; this matches the expected brainwaves for a sleeping individual [51, 52]. The results in Fig. 28 were taken from a study comparing healthy individuals to those suffering from epilepsy [53]. This data, taken from a healthy subject with the eyes open, shows a large portion of the EEG signal in the beta range. This frequency analysis is consistent with the expected results for a conscious indi-

vidual with his or her eyes open. Such results gave us confidence that our FFT algorithm was functioning correctly.

4.4 Human Testing

Skin Preparation and Electrode Placement. We primarily tested this device on Subject A, a 21-year-old male. He completed a skin preparation regimen prior to each test. This regimen consisted of cleaning the skin with an alcohol swab, followed by light abrasion with 220-grit sandpaper. Subject A then put on the headband with the mounted device oriented such that the electrodes rested near the forehead. Next, Ten20 EEG Paste was used to fill the device's gold cup electrodes. Two electrodes were applied on the forehead at the Fp1 and Fp2 positions. The remaining right-leg driver electrode was placed on the subject's right mastoid process, the bone behind the right ear.

Measuring Brainwaves. Signals picked up by the electrodes were fed into the circuitry discussed in section 3.1. The output was connected to an oscilloscope, as well as to the Arduino's analog 1 channel. Figure 30 demonstrates the ability to read the brain's bioelectric signal on an oscilloscope; Figure 31 shows that the same signal can be accurately digitized by the Arduino microcontroller for subsequent analysis. Subject A's EEG was generally found to have an amplitude of 200 mV.

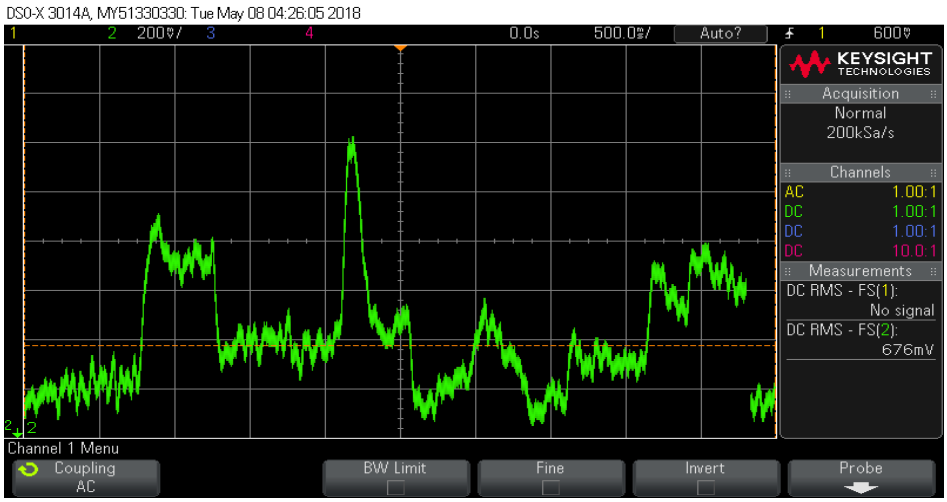


Fig. 29. Oscilloscope Display of Subject A's Brain Signal.

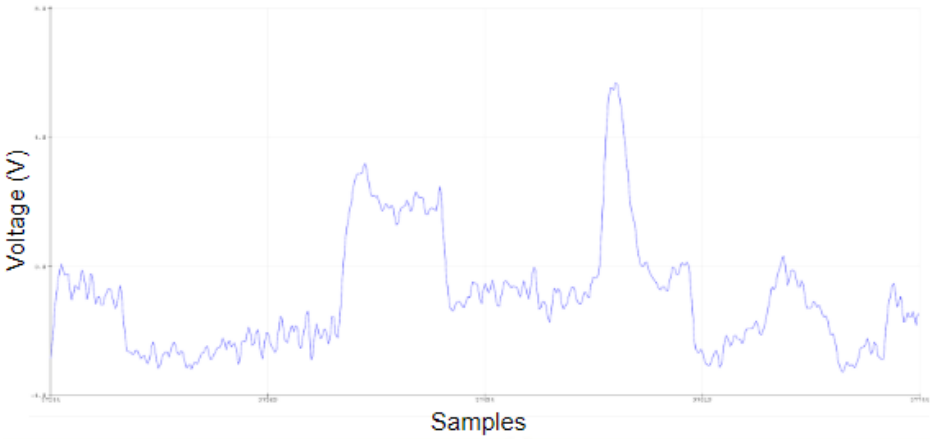


Fig. 30. Arduino Display of Subject A's Brain Signal.

FFT Analysis. One method of analysis subsequent to brainwave collection was putting the digitized brainwave through the arduinoFFT library algorithm and calculating the percentage of the total signal power each brainwave frequency range contained. Figure 31 shows Subject A's brain signal with his eyes closed. Figure 32 gives the resulting FFT analysis of the incoming signal, which indicates a peak in alpha waves. This is consistent with brain waves typically emitted by a conscious individual with their eyes closed.

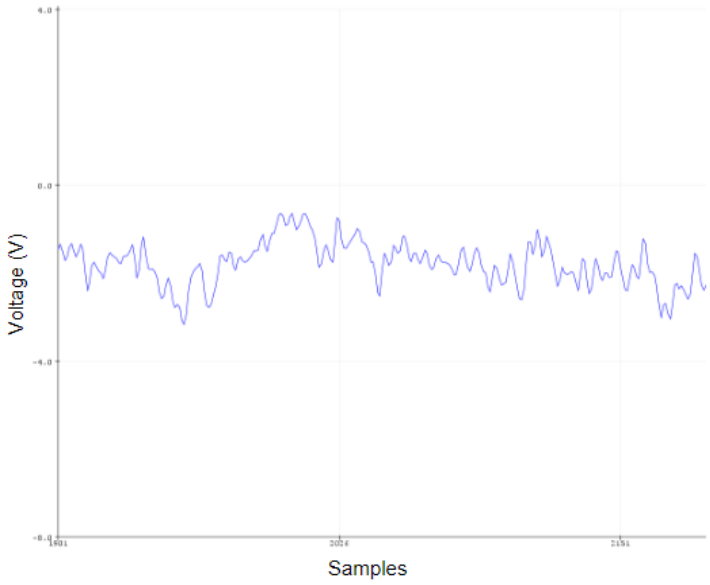


Fig. 31. Subject A's EEG with Eyes Closed.

.

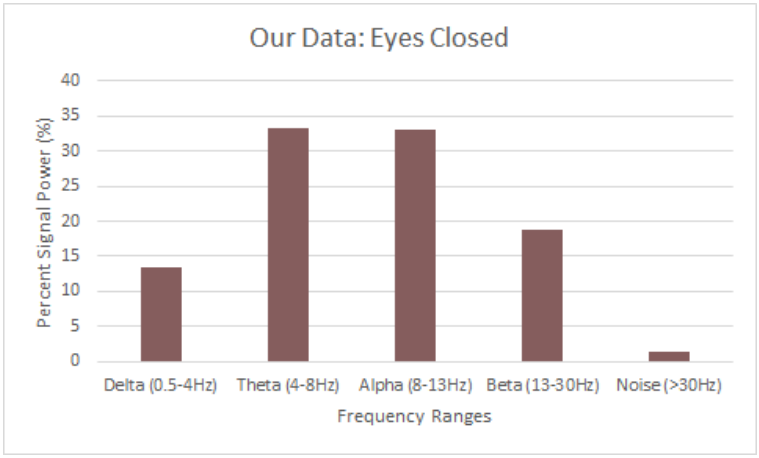


Fig. 32. FFT Analysis of Subject A with Eyes Closed.

Beyond the ability to determine the signal power of each frequency range of Subject A's EEG, we wanted to use such findings to provide audiovisual feedback in real-time. In order to do so, we ran a full system test, using the code in Appendix A-5 to turn on LEDs and a buzzer for eyes open (beta) versus eyes closed (alpha) feedback. With a completed, functional prototype, we revisited our speed and accuracy design specifications to see how well we had hit our targets.

With respect to speed, the system was a bit slower in detecting frequency changes than anticipated—it took around 5 seconds, 4 seconds longer than desired. This could be due to several factors, including residual noise, the Arduino baud rate, and the FFT algorithm itself. A tradeoff exists between FFT speed and resolution (Eq. 3.5 & 3.6). Because we achieved a frequency resolution of 0.78Hz, which was better than our desired specification of 1Hz, the speed of our feedback was negatively impacted.

Band-Pass Filter Analysis. Noticing some inconsistencies between initial human testing and subsequent sessions on Subject A, in which the FFT algorithm outputted high percentages of low frequency delta waves, we designed the digital band-pass filters (BPFs) in Fig. 33 to examine the frequencies of the subject's brainwaves directly.

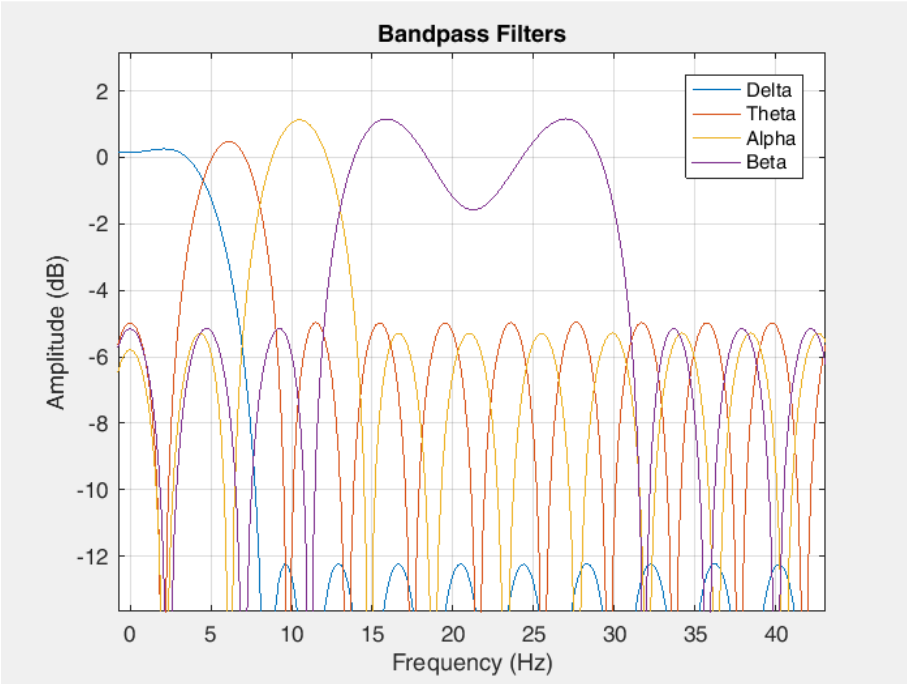


Fig. 33. Brainwave Frequency Range BPF Design.

We applied these individual BPFs to the input signal and simultaneously plotted their outputs. By doing this, we verified the same results the FFT has shown—an abundance of delta waves (Fig. 34).

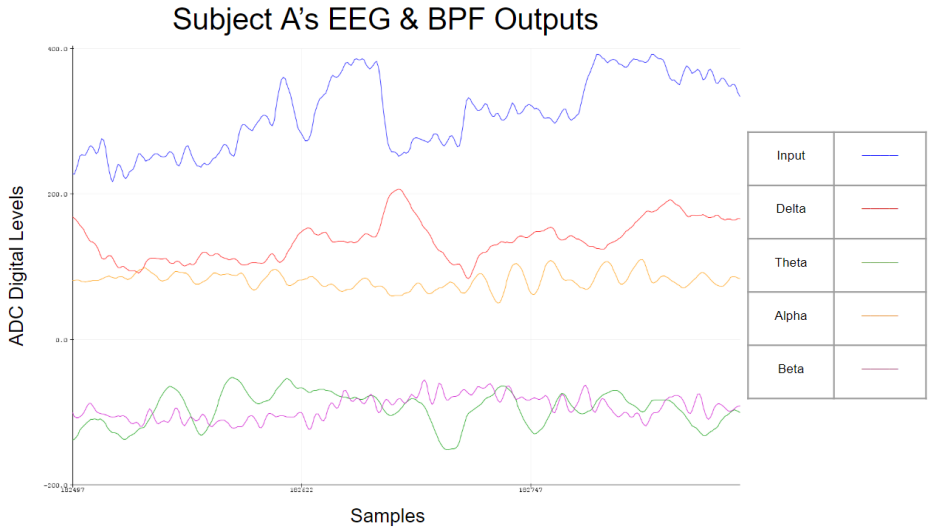


Fig. 34. BPF Output Showing High Amplitude Delta & Theta.

Qualitative Analysis using Band-Pass Filters. In addition to quantitative analysis of Subject A's brainwaves during initial human testing with the FFT, we were able to examine his response to different external stimuli using BPF analysis. Table 6 provides a legend to the input and digitally filtered signals plotted simultaneously in Fig. 35 and Fig. 36. In Fig. 35, Subject A watched a compilation of comedic videos while lying down. It can be seen that the subject's EEG remained both steady and low amplitude with this stimulus. By contrast, Fig. 36 taken in the same manner but with a sad video stimulus, shows shifting brainwaves and large amplitude swings. From these findings, it is evident that our device can detect changes in a subject's mental state or brain activity.

Table 6. Signal legend.

Input Signal	**BLUE**
Delta	**RED**
Theta	**GREEN**
Alpha	**YELLOW**
Beta	**PURPLE**

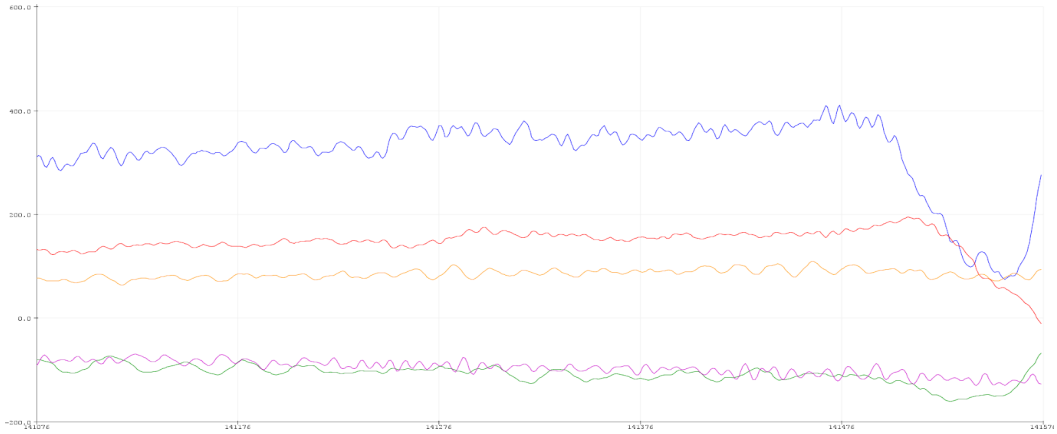


Fig. 35. Subject A's EEG While Watching a Funny Video.

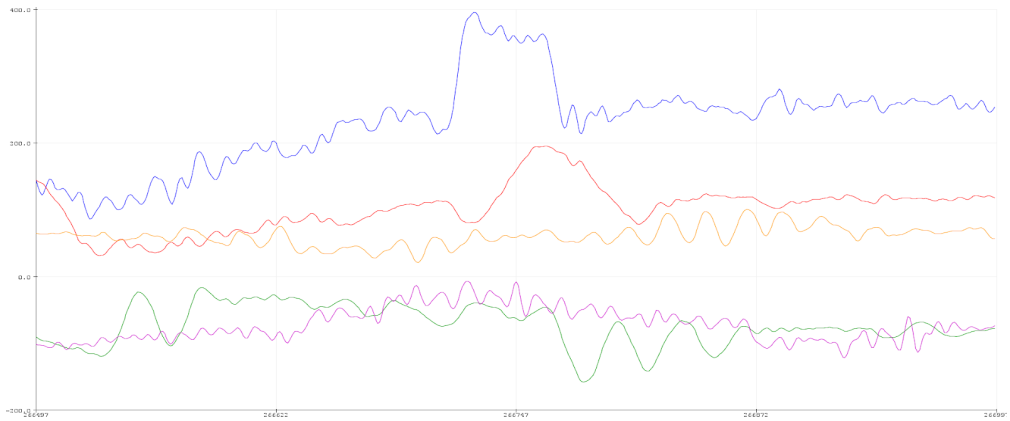


Fig. 36. Subject A's EEG While Watching a Sad Video.

4.5 Discussion

As mentioned in the previous section, after initial testing, the bulk of Subject A's EEG results contained a substantial portion of delta waves in both FFT and BPF analysis. This profusion of low frequencies prevented either analysis method from

acquiring reliable data, which should have been predominantly in the alpha and beta range when the subject's eyes were closed or open, respectively. These low frequencies can be attributed to signal drift, when the center of the signal shifts up or down, and low frequency artifacts from breathing and other bodily movements (Fig. 37). In particular, ocular artifact interference has been found to cause EEG baseline drift [54]. This degradation in performance after primary testing was likely due to poor skin contact caused by electrode deterioration. In fact, during the final rounds of testing, the system was unable to detect an EEG signal with initial electrode placement. It was only after repeating the skin preparation regimen and cleaning and reapplying the electrodes that an EEG signal was found.

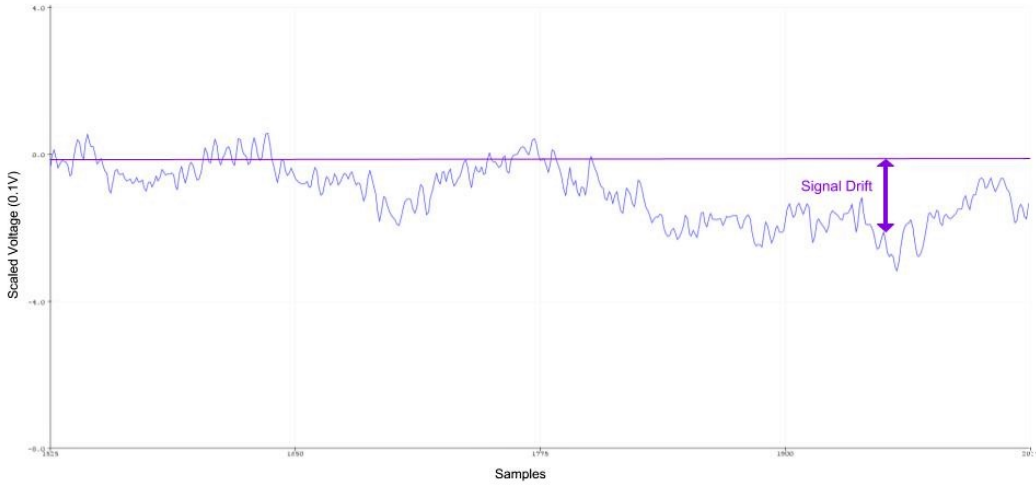


Fig. 37. Low Frequency Signal Drift.

These reliability issues could be addressed in future work. To mitigate electrode deterioration, they could be cleaned more thoroughly, according to manufacturer's instructions after each use. Setting clear user guidelines on the recommended number of uses before electrode replacement is required would also be helpful for the finished product. In addition to these solutions for signal drift, low-frequency noise artifacts could be mitigated through additional hardware components that measure and capture the source of these artifacts. For example, simultaneously measuring heartbeat with an electrocardiogram (ECG) can be used to successfully eliminate ECG artifacts from

raw EEG data, thus greatly improving signal integrity [55]. Software processing through a computational method known as independent component analysis (ICA) in combination with approximations of the surface Laplacian, have also been shown to eliminate contamination of EEG due to electromyogram (EMG) body artifacts [56].

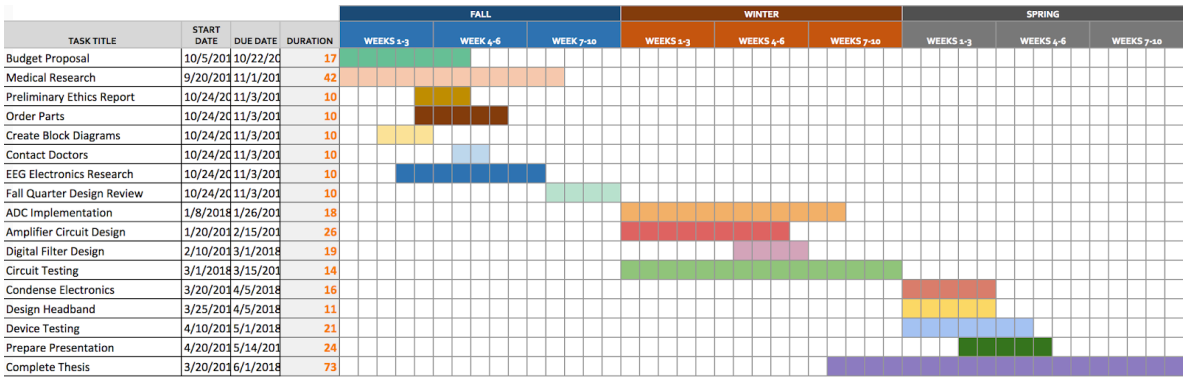
5 Project Details

5.1 Budget

Our budget for this project was $833, which was generously provided by Santa Clara University's School of Engineering.

5.2 Timeline

The Gantt chart in Figure 38 shows a roadmap for the development and testing of our device.



TASK TITLE	START DATE	DUE DATE	DURATION
Budget Proposal	10/5/201	10/22/20	17
Medical Research	9/20/201	11/1/201	42
Preliminary Ethics Report	10/24/20	11/3/201	10
Order Parts	10/24/20	11/3/201	10
Create Block Diagrams	10/24/20	11/3/201	10
Contact Doctors	10/24/20	11/3/201	10
EEG Electronics Research	10/24/20	11/3/201	10
Fall Quarter Design Review	10/24/20	11/3/201	10
ADC Implementation	1/8/2018	1/26/201	18
Amplifier Circuit Design	1/20/201	2/15/201	26
Digital Filter Design	2/10/201	3/1/2018	19
Circuit Testing	3/1/2018	3/15/201	14
Condense Electronics	3/20/201	4/5/2018	16
Design Headband	3/25/201	4/5/2018	11
Device Testing	4/10/201	5/1/2018	21
Prepare Presentation	4/20/201	5/14/201	24
Complete Thesis	3/20/201	6/1/2018	73

Fig. 38. Gantt chart.

5.3 Risk Analysis

There were certain risks associated with the development of this product, especially given the 9-month time limit for the project. Table 7 lists several risks, their probability of occurring, and the impact on the project overall.

Table 7. Summary of risks.

Risk	Probability	Impact
Not obtaining a good EEG signal.	*Medium.* EEG waves have a very low amplitude (100μV).	*High.* Device cannot function without a clean input.
Not finishing the code.	*Low/Medium.* We haven't begun the audiovisual component.	*Medium/High.* Fairly integral to the project.
Not creating a finished headband (ran out of time).	*Medium.* This is our lowest-priority requirement.	*Low.* The device will still function without the headband.
Cost is more than expected.	*Low.* This has been a very low-cost device so far.	*High.* One of our requirements for the device is affordability.

In the end, these risks did not materialize into serious obstacles. However, certain improvements could be made to the device—these will be discussed in the following chapter.

5.4 Bill of Materials

Table 8 shows the bill of materials for the device.

Table 8. Bill of materials.

Item	Cost
Arduino Mega2560	$38.50
USB 2.0 Cable Type A/B	$3.95
TI INA-826 EVM	$25.00
Filter Board	$2.25
Cables	$3.00
Box Construction	$8.00
PCB	$33.00
Surface mount parts	$5.00
Headband	$8.00
Total	$126.70

The overall materials cost was $126.70. However, this cost would significantly decrease if the amplifier and filter boards were condensed into a single PCB of our own design and would be further driven down with mass manufacturing of the device.

6 Professional Issues

6.1 Health and Safety

A pertinent concern with this project is its effect on health and safety. Because the device will be used for therapeutic purposes, it must be designed with the user's health as the highest priority. This follows the cardinal rule in IEEE's code of ethics: "to hold paramount the safety, health, and welfare of the public" [57]. For our project, the health of the user involves many different considerations. The device should not physically harm the patient; the audiovisual feedback should not cause mental stress or pain; the portrayal of data and progress should be honest; and the user interface should not be frustrating to use.

6.2 Usability

Because this product is intended for consumer use (as opposed to doctors and care providers), usability is as important as the technical design. A product with high usability decreases the cognitive load on the user, because the method of operating the device successfully will be apparent in its design.

There are certain design elements within our product that support usability. For example, an on-off switch is used to toggle the power of the device, which many users would be familiar with in other devices. Because our device is mounted on an elastic headband, most users would understand that the device must be worn on the head. The use of an audio jack to connect the electrodes to the device is an affordance that eliminates doubt and confusion.

However, certain improvements to the product's usability could be made. One would be implementing a way to secure the electrodes to the user's forehead without the use of disposable tape. Because the electrodes are used with a slightly sticky, thick electrolyte gel, we are currently holding them in place using tape. Another improvement would be clear labeling of the on/off position on the switch, as well as the audio jacks that correspond to the electrodes and the output. This way, users would spend less time thinking about how to use the device.

6.3 Ethics

There is a clinical need for providing affordable access to neurofeedback for the people who cannot afford the doctor's office visits. But beyond the medical need, providing better access to neurofeedback is an ethical imperative. According to a utilitarian ethics, the most ethical action is the one that increases the sum total of all happiness. A life unburdened by mental health issues is certainly happier than one crippled by depression, anxiety, and other problems; as John Stuart Mill wrote, the highest form of human happiness preserves "a sense of dignity, which all human beings possess in one form or other" [58]. Mental health problems affect one's relationships, career, goals, moods, and daily life, eroding at that sense of dignity. Because neurofeedback may provide a way to relieve those burdens, it helps promote happiness for those suffering from mental health problems. The prohibitive cost of the treatment stands in direct opposition to those ends, so a low-cost form of neurofeedback carries strong ethical justification.

To design an ethical technology, transparency about the security and privacy of user data needs to be at the core of the product. Users should have full access to their information, control the information, and understand what the information means. They should have options for sharing their data and expect a certain degree of privacy.

Designing our product to meet these ethical standards involves many different considerations. In terms of privacy, controlling access to information needs to be built into the software through encryption schemes and password protection. However, we do acknowledge that our device is not foolproof to hacking, which the user would be made aware of. Furthermore, providing users with autonomy means that they actually understand what the data means and how it could be interpreted. By creating a simple, learnable user interface and using plain language, our product would help users understand the nature of their stored data. A final consideration involves the fact that the end user may be a child, especially if the product is used to treat ADHD. Because children do not enjoy the same rights as adults, nor do they have the same amount of autonomy, they would not retain as much control over their data. The product will need to be technologically flexible enough to accommodate these different user needs.

6.4 Sustainability

The physical resources needed to create the device for our project include conductive gel, disposable gel electrodes, disposable AAA batteries, an instrumentation amplifier, a filter board, a microcontroller, and a user interface. In addition to raw materials, the manufacture and assembly of these components involves lead solder, flux, and additional wiring. Equipment involved in assembly and testing includes a soldering iron, a function generator, an oscilloscope, and a laptop to run the microcontroller. Everything runs on electricity. Final operation of the current design would be battery-powered. The current design has a large environmental impact, as the batteries would have to continually be replaced for viable operation. The product life would initially be 30 to 50 hours of use, with an expected total lifespan for the device of 5 to 10 years depending on usage frequency. If the electronics are disposed of responsibly, most of the raw materials of the device can be recycled. Alternatively, they could end up as polluting e-waste.

The electrodes are reusable gold cup electrodes, which are a more sustainable option than disposable gel electrodes. Because the electrolyte gel used with the electrodes is mainly a combination of water and electrolytes, it should be almost 100% biodegradable. This is important since gel must be applied to the skin each time the device is used.

In the future, time permitting, we would like to design a board to power and recharge our device. This design would be more sustainable, since the battery would need to be replaced much less frequently. It may also be possible to redesign the system to work with permanent dry electrodes, even with their lower conductivity.

6.5 Science, Technology, and Society

Mental health problems and behavioral disorders are becoming more prevalent in the U.S. and worldwide (Fig. 39). These issues are especially prevalent among youth—the future of society—leading to juvenile delinquency, dropping out of school, and a less educated population. Those with mental health disorders are also more prone to suicide, which is the third leading cause of death in youth aged 10-24. In addition to the debilitating effects mental health can have on an individual, they have a clear and negative impact on society as a whole. Depression is the leading cause of disability globally, and in the U.S alone, poor mental health contributes to an annual loss of approximately $200 billion in lost productivity [2].

As a growing public health issue, society has begun to recognize mental health and behavioral disorders as real diseases deserving of attention. This has resulted in significantly more resources being devoted to the study of these disorders, and both scientific and technological development working towards their resolution. One such example is Shouse and Lubar's paper first proposing neurofeedback for treatment of ADHD in 1976 and the influx of research and publications on neurofeedback therapy since [59]. This in turn has led to consumer products such as the Neuroband Plus, which, while not being medical-grade, uses the same underlying principles of that

initial scientific research. Our device seeks to further the discoveries of the scientific community by increasing accessibility and ease of use of this novel therapy to benefit the health of society as a whole.

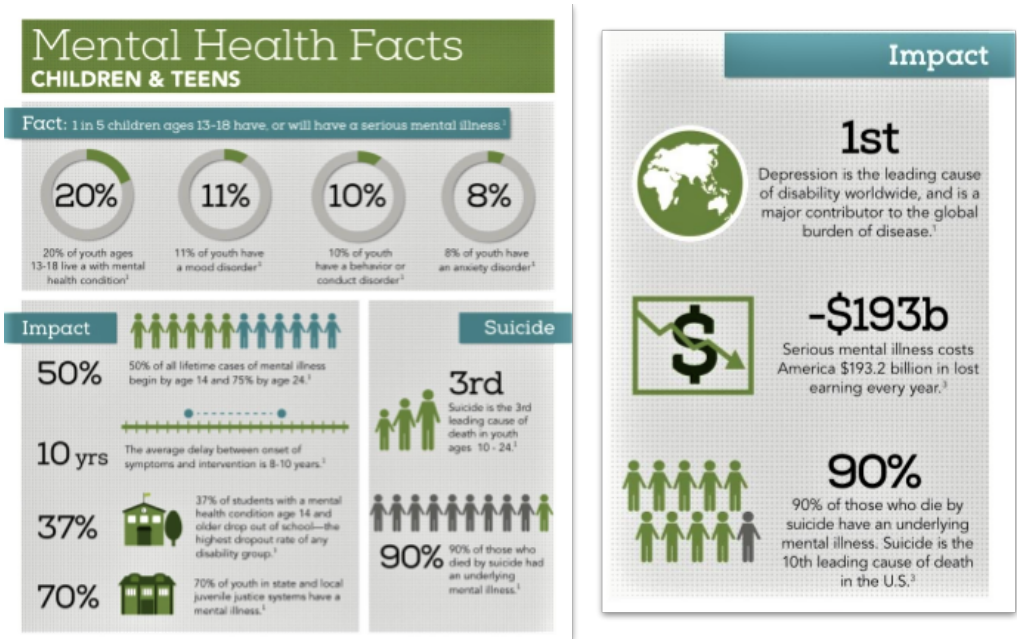


Fig. 39. Mental Health Infographic [2].

6.6 Civic Engagement

The duty of the United States Food & Drug Administration (FDA) is to advance and protect public health "by ensuring the safety, efficacy, and security" of various products, including medical devices [61]. The FDA classifies Neurological Therapeutic Devices as Class II medical devices exempt from premarket approval [61]. In order to seek FDA approval for our device, we would be required to submit a 501(k) premarket submission proving substantial equivalence to another legally U.S. marketed device. Here, substantial equivalence is defined as the new device being "at least as safe and effective as the predicate" [62]. While we have taken steps to influence such approval by using medical grade electrodes and conductive gel, future work would re-

quire design improvements to compensate for electrode degradation and low frequency artifacts, and to improve reliability and efficacy to the necessary FDA level.

7 Conclusion

7.1 Future Work

Several improvements to the device can be made to enhance its usability and performance. One change is printing our own PCB instead of using the evaluation boards manufactured by Texas Instruments and Analog Devices. This would dramatically decrease the size of the circuit and make the device both lighter and smaller. It would also cut costs, since bulk orders of a custom PCB are cheaper than buying each individual evaluation board.

Another improvement would be making the device rechargeable, since it currently runs on four AAA-batteries. This would require another external circuit involving a rechargeable battery and boost converter. With a rechargeable device, users would be able to more conveniently and reliably power the device.

A final but critical improvement involves the feedback component of the system. Currently, we have implemented a simplified audiovisual feedback that consists of colored LEDs and a piezoelectric speaker. A more robust and engaging form of feedback would be a software program that is customizable to the user.

7.2 Summary

The photograph in Figure 40 shows the final prototype and how it would be worn by a user.

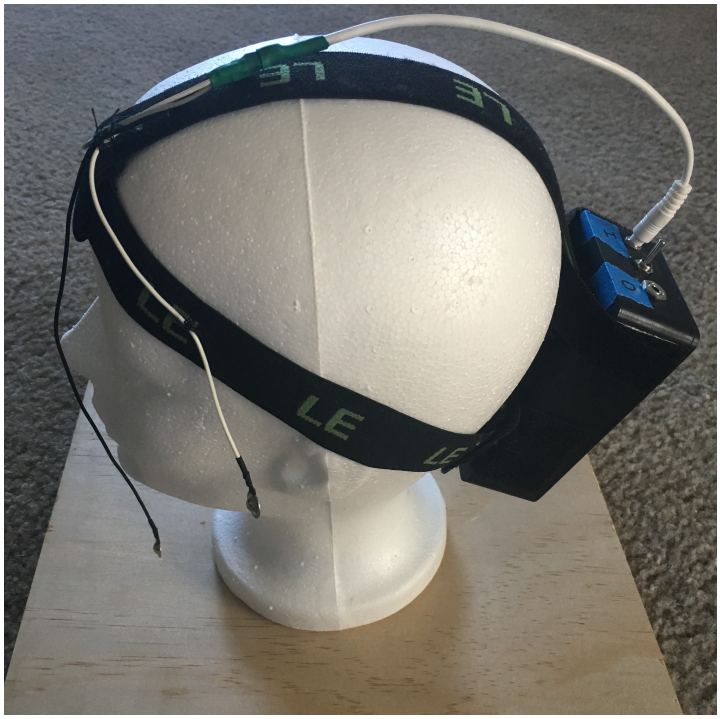


Fig. 40. Final prototype.

A device has been presented that has the potential to treat many brain-related conditions in a non-invasive, non-pharmacologic way. The system allows a user to program the software for his or her particular therapeutic needs and use the principles of neurofeedback to self-regulate brain function. While several improvements to the system could be made, it paves the way for devices that address the important clinical need of solutions for mental health issues.

References

[1] R. Whitaker, "Anatomy of an Epidemic: Psychiatric Drugs and the Astonishing Rise of Mental Illness in America," *Ethical Human Psychology and Psychiatry,* vol. 7, no. 1, pp. 23-35, 2005.

[2] National Alliance on Mental Illness, "NAMI," *Home*. [Online]. Available: https://www.nami.org/Learn-More/Fact-Sheet-Library. [Accessed: 01-Jun-2018].

[3] T. R. Insel, "Assessing the Economic Costs of Serious Mental Illness," *Am J Psychiatry*, vol. 165, no. 6, pp. 663-665, Jun. 2008.

[4] L. A. Pratt, D. J. Brody, and Q. Gu, "Antidepressant Use Among Persons Aged 12 and Over: United States, 2011–2014," *National Center for Health Statistics,* no. 283, Aug. 2017. Available: https://www.cdc.gov/nchs/products/databriefs/db283.htm.

[5] "Neurofeedback and Biofeedback for Mood and Anxiety Disorders: A Review of the Clinical Evidence and Guidelines – An Update," *Canadian Agency for Drugs and Technologies in Health,* Aug. 26 2014.

[6] B. L. Smith, "Inappropriate Prescribing," *American Psychological Association*, vol. 43, no. 6, p. 36, 2012.

[7] J. H. Gruzelier, "EEG-neurofeedback for optimising performance. I: A review of cognitive and affective outcome in healthy participants," *Neurosci Biobehav Rev*, vol. 44, pp. 124-141, Jul. 2014.

[8] T. Fuchs, et. al, "Neurofeedback Treatment for Attention-Deficit/Hyperactivity Disorder in Children: A Comparison with Methylphenidate," *Applied Psychophysiology and Biofeedback,* vol. 28, no. 1, pp. 1-12, Mar. 2003.

[9] R. Sitaram, et. al, "Closed loop brain training: The science of neurofeedback," *Nature*, vol. 18, pp. 86-100, Feb. 2017.

[10] K. Fox and M. Stryker, "Integrating Hebbian and homeostatic plasticity: introduction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 372, Mar. 2017.

[11] R. J. Schafer, et al, "Selective attention from voluntary control of neurons in prefrontal cortex," *Science*, vol. 332, pp. 1568–1571, 2011.

[12] A. C. Koralek, et al, "Corticostriatal plasticity is necessary for learning intentional neuroprosthetic skills," *Nature,* vol. 483, pp. 331-335, Mar. 2012.

[13] T. Ros, et. al, "Endogenous control of waking brain rhythms induces neuroplasticity in humans," *European Journal of Neuroscience*, vol. 31, pp. 770-778, Feb. 2010.

[14] M. Teplan, "Fundamentals of EEG Measurement," *Measurement Science Review,* vol. 2, no. 2, pp. 1-11, 2002.

[15] "10-20 System (EEG)," *Wikipedia,* Mar. 11 2018. Available: https://en.wikipedia.org/wiki/10%E2%80%9320_system_(EEG). [Accessed: May 3, 2018].

[16] M. Abo-Zahhad, S. M. Ahmed, and S. N. A. Seha, "A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 6, pp. 48-54, May 2015.

[17] W. Klonowski, "Everything You Wanted to Ask about EEG but Were Afraid to Get the Right Answer," *Nonlinear Biomed Phys,* vol. 3, p. 2, May 2018.

[18] H. Marzbani, H. R. Marateb, and M. Mansourian, "Neurofeedback: a comprehensive review on system design, methodology and clinical applications," *Basic and Clinical Neuroscience*, vol 7, no. 2, pp. 143-158, 2016.

[19] J. L. Cantero, M. Atienza, and R. M. Salas, "Human alpha oscillations in wakefulness, drowsiness period, and REM sleep: different electroencephalographic phenomena within the alpha band," *Clin Neurophysiol*, vol. 32, no. 1, pp. 54-71, 2002.

[20] A. Fink and M. Benedek, "EEG alpha power and creative ideation," *Neurosci Biobehav Rev*, vol. 44, pp. 111-123, Jul. 2014.

[21] E. Niedermeyer, "Alpha rhythms as physiological and abnormal phenomena," Int J of Psychophysiol, vol. 26, no. 1–3, pp. 31-49, Jun. 1997.

[22] W. Klimesch, "EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis," *Brain Res Rev*, vol. 29, no. 2–3, pp. 169-195, Apr. 1999.
[23] J. Winson, "Loss of hippocampal theta rhythm results in spatial memory deficit in the rat," *Science*, vol. 201, no. 4351, pp. 160-163, Jul. 1978.
[24] P. Maquet, et. al, "Functional Neuroanatomy of Human Slow Wave Sleep," *Journal of Neuroscience*, vol. 17, no. 8, pp. 2807-2812, 1997.
[25] I. M. Colrain, et. al, "Sleep evoked delta frequency responses show a linear decline in amplitude across the adult lifespan," *Neurobiol Aging,* vol. 31, pp. 874-883, May 2010.
[26] C. H. Schenck, et. al, "Analysis of Polysomnographic Events Surrounding 252 Slow-Wave Sleep Arousals in Thirty-Eight Adults With Injurious Sleepwalking and Sleep Terrors," *Jour. of Clin. Neurophysiology,* vol. 15*,* pp. 159-166, Mar. 1998.
[27] M. Arns, H. Heinrich, and U. Strehl, "Evaluation of neurofeedback in ADHD: The long and winding road," *Biol Psych,* vol. 95, pp. 108-115, Jan. 2014.
[28] D. C. Hammond, "Neurofeedback Treatment of Depression and Anxiety," *Jour. of Adult Development,* vol. 12, no. 2, pp. 131-137, Aug. 2005.
[29] E. Baehr, JP Rosenfeld, and R. Baehr, "The Clinical Use of An Alpha Asymmetry Protocol in the Neurofeedback Treatment of Depression," *Journal of Neurotherapy,* vol. 2, no. 3, pp. 10-23, 2008.
[30] L. Thompson, M. Thompson, and A. Reid, "Neurofeedback outcomes in clients with Asperger's syndrome," *Appl Psychophysiol Biofeedback,* vol. 35, no. 1, pp. 63-81, Mar. 2010.
[31] J. Hamilton, "Train The Brain: Using Neurofeedback To Treat ADHD," *NPR*, Nov. 1 2010. Available: https://www.npr.org/templates/story/story.php?storyId=130896102.
[32] "Muse: The Brain Sensing Headband," *The Gadget Flow.* Accessed: https://thegadgetflow.com/portfolio/muse-the-brain-sensing-headband/. [Accessed: Jun. 1, 2018].
[33] E. Strickland, "Brain-Controlled Game Helps Kids With ADHD Improve Mental Focus," *IEEE Spectrum*, Sep. 26 2017. Available: https://spectrum.ieee.org/the-human-os/biomedical/devices/brain-controlled-game-helps-kids-with-adhd-improve-mental-focus.
[34] "NeuroPlus Focus-Improving Headset," *The Gadget Flow*. Available: https://thegadgetflow.com/portfolio/focus-improving-headset/. [Accessed: Jun. 1, 2018].
[35] "Emotiv Insight 5 Channel Mobile EEG," *Emotiv*, 2018. Available: https://www.emotiv.com/product/emotiv-insight-5-channel-mobile-eeg/. [Accessed: 6/1/18].
[36] P. Tallgren, S. Vanhatalo, K. Kaila, and J. Voipio, "Evaluation of commercially available electrodes and gels for recording of slow EEG potentials," *Clin. Neurophysiol.,* vol. 116, pp. 799-806, 2005.
[37] A. Tautan, V. Mihajlović, C. Y. Hsuan, and W. A. Serdijn, "Signal Quality in Dry Electrode EEG and the Relation to Skin-electrode Contact Impedance Magnitude," In Proc. 7th International Conference on Biomedical Electronics and Devices, 2014.
[38] A. C. Metting van Rijn, A. Peper, and C. A. Grimbergen, "High-quality recording of bioelectric events," *Med & Biol Eng & Comput,* vol. 28, pp. 389-397, 1990.
[39] B. B. Winter and J. G. Webster, "Driven-right-leg circuit design," *IEEE Transactions on Biomedical Engineering*, vol. 30, no. 1, pp. 62-66, 1983.
[40] T. S. Ruby, "Audio Detection Circuit," *Farhek,* Jan. 21, 2017. Available: http://farhek.com/pt/mf12166/detection-circuit/. [Accessed: May 28, 2018].
[41] Texas Instruments, "User's Guide," INA826EVM datasheet, Aug. 2011 [Revised Mar. 2012].
[42] Texas Instruments, "Instrumentation Amplifier with Rail-to-Rail Output," INA826 datasheet, Aug. 2011 [Revised Jul. 2016].
[43] "A Beginner's Guide to Filter Topologies," *Maxim Integrated,* Feb. 4, 2003. Available: https://www.maximintegrated.com/en/app-notes/index.mvp/id/1762. [Accessed: May 23, 2018].
[44] X. Qian, Y. P. Xu, and X. Li, "A CMOS Continuous-Time Low-Pass Notch Filter for EEG Systems. Analog INtegrated Circuits and Signal Processing," vol. 44, pp. 231-238, 2005.

[45] "Notch Filter," *Learning about Electronics*. Available: http://www.learningaboutelectronics.com/Articles/Notch-filter-calculator.php. [Accessed: May 23, 2018].
[46] Need
[47] I. Poole, "Op Amp Notch Filter Circuit," *Radio Electronics*. [Online]. Available: http://www.radio-electronics.com/info/circuits/opamp_notch_filter/opamp_notch_filter.php. [Accessed: May 10, 2018].
[48] "High Q Notch Filter," *National Semiconductor Corporation,* 1995. [Online]. Available: http://g3ynh.info/circuits/Bob_Batey/LB-5.pdf. [Accessed: May 25, 2018].
[49] Kosme, "kosme/arduinoFFT," *GitHub*, 11-Feb-2018. [Online]. Available: https://github.com/kosme/arduinoFFT. [Accessed: 11-Jun-2018].
[50] R. Nave, "Square Wave Generator," *HyperPhysics.* [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/square.html. [Accessed: May 20, 2018].
[51] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberye, "Analysis of sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185-1194, 2000.
[52] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. 215–220, Jun. 2000.
[53] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, vol. 64, no. 6, Nov. 2001.
[54] D. Cuesta-Frau, P. Miró-Martínez, J. J. Núñez, S. Oltra–Crespo, and A. M. Picó, "Noisy EEG signals classification based on entropy metrics. Performance assessment using first and second generation statistics," *Computers in Biology and Medicine*, vol. 87, pp. 141–151, 2017.
[55] Y. Okamoto, M. Yuki, R. Imai, Y. Shinki, M. Yamanaka, and T. Masutani, "Elimination of electrocardiogram artifacts from electroencephalogram records by using the simultaneously recorded electrocardiograph data," *Rinsho Byori: The Japanese Journal of Clinical Pathology*, vol. 50, no. 12, pp. 1150–1153, Dec. 2002.
[56] S. Fitzgibbon, D. Delosangeles, T. Lewis, D. Powers, E. Whitham, J. Willoughby, and K. Pope, "Surface Laplacian of scalp electrical signals and independent component analysis resolve EMG contamination of electroencephalogram," *International Journal of Psychophysiology*, vol. 97, no. 3, pp. 277–284, 2015.
[57] "IEEE Code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Apr. 10, 2018].
[58] J. S. Mill, *Utilitarianism*, 1861. CreateSpace Independent Publishing Platform, 2017.
[59] J. F. Lubar and M. N. Shouse, "EEG and behavioral changes in a hyperkinetic child concurrent with training of the sensorimotor rhythm (SMR)," *Biofeedback and Self-Regulation*, vol. 1, no. 3, pp. 293–306, 1976.
[60] O. of the Commissioner, "What We Do," *U S Food and Drug Administration Home Page*. [Online]. Available: https://www.fda.gov/AboutFDA/WhatWeDo/. [Accessed: 31-May-2018].
[61] "CFR - Code of Federal Regulations Title 21," *accessdata.fda.gov*, 01-Apr-2017. [Online]. Available: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=882.5050. [Accessed: 31-May-2018].
[62] Center for Devices and Radiological Health, "Premarket Notification 510(k)," *U S Food and Drug Administration Home Page*. [Online]. Available: https://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/HowtoMarketYourDevice/PremarketSubmissions/PremarketNotification510k/default.htm. [Accessed: 31-May-2018].

A Appendix

A-1 Matlab Bandpass Filter Design

```
clearvars;
clc;

Fs = 200;  %Set the sampling rate.
Ts = 1/Fs; %Calculate the sampling period.

%Use the FIRPM algorithm to design the filter.

%N is the number of coefficients the filter will have.
N = 50;

%{
The following F arrays are the critical points in each filter: delta,
theta, alpha, and beta. The points range from 0 to 1. 0 represents
0Hz and 1 represents Fs/2. For example, 30Hz can be represented by
30/(Fs/2) = 0.3.
%}

FD = [0 .04 .08  1];
FT = [0 .03 .04 .08 .09 1];
FA = [0 .07 .08 .13 .14 1];
FB = [0 .12 .13 .3 .31 1];

%{
Arrays A and B below match the size of F arrays listed above and de-
scribe what happens between the critical points listed in each F ar-
ray. 1 represents the signal passing through (pass band) while 0 rep-
resents the signal being blocked (stop band). For FT and the values
of B=[0 0 1 1 0 0], this means 0 to 3 Hz will be blocked by, 4 to 8
Hz will be passed through, and 9 to 100 Hz will be blocked by the
filter. The values between 0.03 and 0.04 as well as between 0.08 and
0.09 are in the transition band. The smaller the transition band, the
larger the filter will need to be to increase rolloff. }%


A = [1 1 0 0 ];
B = [0 0 1 1 0 0];

%Finally, the coefficients for each filter will be returned and
stored in
%each array
BD = transpose(firpm(N,FD,A));
BT = transpose(firpm(N,FT,B));
BA = transpose(firpm(N,FA,B));
BB = transpose(firpm(N,FB,B));

%{
```

```
In order to plot the filter, to verify shape, use freqz(). The first
argument is the filter coefficients. The second are feedback coeffi-
cients, but for an FIR filter that is 1. The third argument repre-
sents the number of points to plot. Finally, 'whole' plots the entire
frequency spectrum symmetrically.
%}

[HA,WA] = freqz(BA,1,1024,'whole');
[HD,WD] = freqz(BD,1,1024,'whole');
[HT,WT] = freqz(BT,1,1024,'whole');
[HB,WB] = freqz(BB,1,1024,'whole');

%{
The returned values from freqz() can be formatted as follows to plot
the magnitude in dB from -Fs/2 to Fs/2 and centered at 0Hz
%}

figure(1);
plot(((WD-pi)/pi)*(Fs/2), 10*log10(fftshift(abs(HD))));
xlabel('Frequency (Hz)');
ylabel('Amplitude (dB)');
title('Bandpass Filters');
grid on;
hold on;
plot(((WT-pi)/pi)*(Fs/2), 10*log10(fftshift(abs(HT))));
plot(((WA-pi)/pi)*(Fs/2), 10*log10(fftshift(abs(HA))));
plot(((WB-pi)/pi)*(Fs/2), 10*log10(fftshift(abs(HB))));
legend('Delta','Theta','Alpha','Beta','Location','northeast')
```

A-2 Matlab Low-Pass Filter Design

```
clearvars;
clc;

Fs = 200;  %Set the sampling rate.
Ts = 1/Fs; %Calculate the sampling period.

%Use the FIRPM algorithm to design the filter.

%N is the number of coefficients the filter will have.
N = 20;

%{
The F array contains the critical points of the filter. The points
range
from 0 to 1. 0 represents 0Hz and 1 represents Fs/2. For example,
50Hz
can be represented by 50/(Fs/2) = 0.5.
%}

F = [0 .3 .5  1];

%{
Array A is the same size as array F listed above and describes what
happens between the critical points listed in the F array. 1 repre-
sents the signal passing through (pass band) while 0 represents the
signal being blocked (stop band). For F and the values of =[1 1 0 0],
this means 0 to 30Hz will be passed through and 50 to 100Hz will be
blocked by the filter.
The values between 0.3 and 0.5 as well are in the transition band.
The smaller the transition band, the larger the filter will need to
be to increase rolloff. %}

A = [1 1 0 0 ];

%Finally, the coefficients for the filter will be returned and stored
in B.

B = transpose(firpm(N,F,A));

%{
In order to plot the filter, to verify shape, use freqz(). The first
argument is the filter coefficients. The second are feedback coeffi-
cients, but for an FIR filter that is 1. The third argument repre-
sents the number of points to plot. Finally, 'whole' plots the entire
frequency spectrum symmetrically.
%}

[H,W] = freqz(B,1,1024,'whole');
```

```
%{
The returned values from freqz() can be formatted as follows to plot
the magnitude in dB from -Fs/2 to Fs/2 and centered at 0Hz
%}
figure(1);
plot(((W-pi)/pi)*(Fs/2), 10*log10(fftshift(abs(H))));
xlabel('Frequency (Hz)');
ylabel('Amplitude (dB)');
title('Lowpass Filters');
grid on;
```

A-3 Arduino Input / Filtered Signal Plotter

```
float readValue;
float out_data;
float voltage;

void setup(){
  // System external reference voltage set to 0.825V
  analogReference(EXTERNAL);
  Serial.begin(9600);
}

//Using the Serial plotter, this plots a real-time graph of the sig-
nal and optionally, the filtered output
void loop() {
    //Call sample function
    sample();

    //Print either analog signal digital levels or voltage
    /* MAKE SURE THIS IS CONSISTENT WITH SELECTION IN sample()
FUNCTION */
    Serial.println(voltage);  //Serial.println(readValue);

    /* Option to print out filtered signal on same plot */
    //Serial.print("  ");
    //Serial.println(out_data,4);
}

//Takes an analog reading and passes it through fir_filt
void sample()
{
  //Read in analog signal, using channel 1
  readValue = analogRead(1);

  //Convert analogRead to voltage, shift signal for DC bias correc-
tion, scale by 10 for visualization
  voltage = 10*((readValue * (0.825/1024)-0.25));

  //Pass input through LPF: can use analog reading directly or calcu-
lated voltage
  fir_filt((float)voltage, &out_data);
}

// Low pass filter function: convolves input with filter coefficients
void fir_filt(float in_data, float* out_data)
{
  static const int N = 21; //Filter length.

  //Filter coefficients.  Only half will be used as the coefficients
are symmetric.
  static const float coeff[] =
```

```
  {
    0.001214735329243, -0.012960952370491, -0.010093833713808,
 0.013957771298692,  0.031889096998368,
    0.000124591983717, -0.062115960863680, -0.056006130074433,
 0.089034472964041,  0.299161329745819,
    0.400142978571672,  0.299161329745819,  0.089034472964041, -
0.056006130074433, -0.062115960863680,
    0.000124591983717,  0.031889096998368,  0.013957771298692, -
0.010093833713808, -0.012960952370491,
    0.001214735329243
  };

  static float shift_reg[N]; //Shift register for storing incoming
data values.

  float acc = 0; //Accumulator.

  //Shift all values.
  for (int i = N - 1; i >= 0; i--)
  {
    shift_reg[i+1] = shift_reg[i];
  }

  shift_reg[0] = in_data; //Get input data.

  //Multiply through. Take advantage of symmetry.
  for (int i = 0; i < N/2; i++)
  {
    acc += coeff[i] * (shift_reg[i] + shift_reg[N - 1 - i]);
  }

  //Need the middle value if there is an odd number of filter taps.
  if(N % 2) acc += shift_reg[N / 2] * coeff[N / 2];

  *out_data = acc; //Output result.
}
```

A-4 Arduino BPF Test Code

```
int readValue;
float out_data_a; float out_data_b; float out_data_t; float
out_data_d;

void setup() {
    // System external reference voltage set to 0.825V
    analogReference(EXTERNAL);
    Serial.begin(115200);
}

void loop() {
  //Read in analog signal, using channel 1
  readValue =  analogRead(1);

  //Pass input signal through delta, theta, alpha, and beta BPFs
  fir_delta((float)readValue, &out_data_d);
  fir_theta((float)readValue, &out_data_t);
  fir_alpha((float)readValue, &out_data_a);
  fir_beta((float)readValue, &out_data_b);

  //Print input and filtered signals
  Serial.print(readValue);
  Serial.print(" ");
  Serial.print(out_data_d);
  Serial.print(" ");
  Serial.print(out_data_t);
  Serial.print(" ");
  Serial.print(out_data_a);
  Serial.print(" ");
  Serial.println(out_data_b);
}

//Delta BPF function: convolves input with delta filter coefficients
void fir_delta(float in_data, float* out_data_d)
{
  static const int N = 51; //Filter length.

  //Filter coefficients.  Only half will be used as the coefficients
are symmetric.
  static const float coeff[] =
  {
  -0.231989801011412
,0.00642896870693052
,0.00670106121574697
,0.00716253259463093
,0.00774137050523428
,0.00856804703563915
,0.00952664635230532
,0.0106576682928984
,0.0118701899771757
```

```
,0.0132125072320391
,0.0145968980786637
,0.0160892704991578
,0.0175701467750487
,0.0191224885186603
,0.0205561322589260
,0.0219897870291446
,0.0234303961359035
,0.0250666716645814
,0.0263866330550771
,0.0270535014945984
,0.0282008675142371
,0.0288800746523137
,0.0296326137319566
,0.0299857891202791
,0.0303482206539527
,0.0303471886038398
,0.0303482206539527
,0.0299857891202791
,0.0296326137319566
,0.0288800746523137
,0.0282008675142371
,0.0270535014945984
,0.0263866330550771
,0.0250666716645814
,0.0234303961359035
,0.0219897870291446
,0.0205561322589260
,0.0191224885186603
,0.0175701467750487
,0.0160892704991578
,0.0145968980786637
,0.0132125072320391
,0.0118701899771757
,0.0106576682928984
,0.00952664635230532
,0.00856804703563915
,0.00774137050523428
,0.00716253259463093
,0.00670106121574697
,0.00642896870693052
,-0.231989801011412
  };

  static float shift_reg[N]; //Shift register for storing incoming
data values.

  float acc = 0; //Accumulator.

  //Shift all values.
  for (int i = N - 1; i >= 0; i--)
  {
    shift_reg[i+1] = shift_reg[i];
```

```
  }

  shift_reg[0] = in_data; //Get input data.

  //Multiply through. Take advantage of symmetry.
  for (int i = 0; i < N/2; i++)
  {
    acc += coeff[i] * (shift_reg[i] + shift_reg[N - 1 - i]);
  }

  //Need the middle value if there is an odd number of filter taps.
  if(N % 2) acc += shift_reg[N / 2] * coeff[N / 2];

  *out_data_d = acc; //Output result.
}

//Theta BPF function: convolves input with theta filter coefficients
void fir_theta(float in_data, float* out_data_t)
{
  static const int N = 51; //Filter length.

  //Filter coefficients.  Only half will be used as the coefficients
are symmetric.
  static const float coeff[] =
  {
    0.138223056726333
,-0.0402614181999904
,-0.0376021970904836
,-0.0366391730897299
,-0.0369065040545399
,-0.0377542803877460
,-0.0389307225387330
,-0.0397854836865134
,-0.0402821704094474
,-0.0398889568948220
,-0.0386391818901554
,-0.0360456507295390
,-0.0322903103839652
,-0.0272029107530955
,-0.0213049156345884
,-0.0144953984534086
,-0.00663800736381792
,0.00125025596636957
,0.00947301874896787
,0.0173865141294253
,0.0248262293279388
,0.0314179945033858
,0.0367593310146032
,0.0408515800889782
,0.0433123735478888
,0.0442192809597164
,0.0433123735478888
```

```
,0.0408515800889782
,0.0367593310146032
,0.0314179945033858
,0.0248262293279388
,0.0173865141294253
,0.00947301874896787
,0.00125025596636957
,-0.00663800736381792
,-0.0144953984534086
,-0.0213049156345884
,-0.0272029107530955
,-0.0322903103839652
,-0.0360456507295390
,-0.0386391818901554
,-0.0398889568948220
,-0.0402821704094474
,-0.0397854836865134
,-0.0389307225387330
,-0.0377542803877460
,-0.0369065040545399
,-0.0366391730897299
,-0.0376021970904836
,-0.0402614181999904
,0.138223056726333
  };

  static float shift_reg[N]; //Shift register for storing incoming
data values.

  float acc = 0; //Accumulator.

  //Shift all values.
  for (int i = N - 1; i >= 0; i--)
  {
    shift_reg[i+1] = shift_reg[i];
  }

  shift_reg[0] = in_data; //Get input data.

  //Multiply through. Take advantage of symmetry.
  for (int i = 0; i < N/2; i++)
  {
    acc += coeff[i] * (shift_reg[i] + shift_reg[N - 1 - i]);
  }

  //Need the middle value if there is an odd number of filter taps.
  if(N % 2) acc += shift_reg[N / 2] * coeff[N / 2];

  *out_data_t = acc; //Output result.
}

//Alpha BPF function: convolves input with alpha filter coefficients
```

```
void fir_alpha(float in_data, float* out_data_a)
{
  static const int N = 51; //Filter length.

  //Filter coefficients.  Only half will be used as the coefficients
are symmetric.
  static const float coeff[] =
  {
0.0652924919346733
,-0.103051976827293
,-0.00808305384250486
,0.0386852697978632
,0.0569524909916705
,0.0601088087478705
,0.0558888439317188
,0.0479226904468471
,0.0374592935839687
,0.0247169384727528
,0.0101475017638780
,-0.00547877317437946
,-0.0210978701904859
,-0.0352885239596262
,-0.0464152400368263
,-0.0529661687467591
,-0.0539975924766903
,-0.0491360540833116
,-0.0386555442184646
,-0.0234564966586875
,-0.00521960237411641
,0.0139478751866762
,0.0318076486393737
,0.0463886753774022
,0.0559588763678667
,0.0593152028935091
,0.0559588763678667
,0.0463886753774022
,0.0318076486393737
,0.0139478751866762
,-0.00521960237411641
,-0.0234564966586875
,-0.0386555442184646
,-0.0491360540833116
,-0.0539975924766903
,-0.0529661687467591
,-0.0464152400368263
,-0.0352885239596262
,-0.0210978701904859
,-0.00547877317437946
,0.0101475017638780
,0.0247169384727528
,0.0374592935839687
,0.0479226904468471
,0.0558888439317188
```

```
,0.0601088087478705
,0.0569524909916705
,0.0386852697978632
,-0.00808305384250486
,-0.103051976827293
,0.0652924919346733
  };

  static float shift_reg[N]; //Shift register for storing incoming
data values.

  float acc = 0; //Accumulator.

  //Shift all values.
  for (int i = N - 1; i >= 0; i--)
  {
    shift_reg[i+1] = shift_reg[i];
  }

  shift_reg[0] = in_data; //Get input data.

  //Multiply through. Take advantage of symmetry.
  for (int i = 0; i < N/2; i++)
  {
    acc += coeff[i] * (shift_reg[i] + shift_reg[N - 1 - i]);
  }

  //Need the middle value if there is an odd number of filter taps.
  if(N % 2) acc += shift_reg[N / 2] * coeff[N / 2];

  *out_data_a = acc; //Output result.
}

//Beta BPF function: convolves input with beta filter coefficients
void fir_beta(float in_data, float* out_data_b)
{
  static const int N = 51; //Filter length.

  //Filter coefficients.  Only half will be used as the coefficients
are symmetric.
  static const float coeff[] =
  {
    0.0536340464424499
,-0.108600112694148
,0.00766878236906931
,0.0254428259582104
,-0.00677230282777140
,-0.0492815163681410
,-0.0741539515728317
,-0.0678979012661514
,-0.0333650646408177
,0.0114952981826802
```

```
,0.0432646722257317
,0.0474897827850052
,0.0278395682059267
,0.00378570946606182
,-0.00318569305267785
,0.0130942726737889
,0.0363899185847096
,0.0387803536494570
,0.00236825770767809
,-0.0632963472696998
,-0.121628679324981
,-0.130207737396317
,-0.0702094941907102
,0.0368703738060489
,0.138149940074226
,0.179486208005455
,0.138149940074226
,0.0368703738060489
,-0.0702094941907102
,-0.130207737396317
,-0.121628679324981
,-0.0632963472696998
,0.00236825770767809
,0.0387803536494570
,0.0363899185847096
,0.0130942726737889
,-0.00318569305267785
,0.00378570946606182
,0.0278395682059267
,0.0474897827850052
,0.0432646722257317
,0.0114952981826802
,-0.0333650646408177
,-0.0678979012661514
,-0.0741539515728317
,-0.0492815163681410
,-0.00677230282777140
,0.0254428259582104
,0.00766878236906931
,-0.108600112694148
,0.0536340464424499
  };

  static float shift_reg[N]; //Shift register for storing incoming
data values.

  float acc = 0; //Accumulator.

  //Shift all values.
  for (int i = N - 1; i >= 0; i--)
  {
    shift_reg[i+1] = shift_reg[i];
  }
```

```
  shift_reg[0] = in_data; //Get input data.

  //Multiply through. Take advantage of symmetry.
  for (int i = 0; i < N/2; i++)
  {
    acc += coeff[i] * (shift_reg[i] + shift_reg[N - 1 - i]);
  }

  //Need the middle value if there is an odd number of filter taps.
  if(N % 2) acc += shift_reg[N / 2] * coeff[N / 2];

  *out_data_b = acc; //Output result.
}
```

A-5 Arduino Oscillator Circuit Demonstration

```
#include "arduinoFFT.h"

#define SAMPLES 256             //Must be a power of 2
#define SAMPLING_FREQUENCY 200 //Hz, must be less than 10000 due to
ADC

arduinoFFT FFT = arduinoFFT();

unsigned int sampling_period_us;
unsigned long microseconds;

double vReal[SAMPLES];
double vImag[SAMPLES];

#define LED_T 22
#define LED_B 24

void setup() {
    // System external reference voltage set to 0.825V
    analogReference(EXTERNAL);
    Serial.begin(9600);

    // Set pins corresponding to negative/positive feedback to be
outputs
    pinMode(LED_T, OUTPUT);
    pinMode(LED_B, OUTPUT);

    // Calculate sampling period in microseconds
    sampling_period_us = round(1000000*(1.0/SAMPLING_FREQUENCY));
}

void loop() {

    /*SAMPLING*/
    for(int i=0; i<SAMPLES; i++)
    {
        microseconds = micros();    //Overflows after around 70 min-
utes!

        vReal[i] = analogRead(1);
        vImag[i] = 0;

        while(micros() < (microseconds + sampling_period_us)){
        }
    }

    /*FFT*/
    FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
    FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
    FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);
```

```
    //Find the signal's peak frequency from FFT analysis
    double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING_FREQUENCY);

    /*PRINT RESULTS*/
    Serial.println("Peak = " + String(peak));     //Print out what
frequency is the most dominant.

    /*LED Feedback:
    This demo is set for treatment of ADHD, where an increase in beta
(13-30Hz) and decrease in
    theta (4-8Hz) is desired. A tolerance of 0.8Hz was added to each
frequency range based on
    FFT resolution.*/
    //If the peak falls in theta, turn on LED_T. This is a red LED
for negative feedback.
    if(peak >= 3.2 && peak <= 8.8)
    {
      digitalWrite(LED_T, HIGH);
    }
    else
    {
      digitalWrite(LED_T, LOW);
    }

    //If the peak falls in beta, turn on LED_B. This is a green LED
for positive feedback.
    if(peak >= 12.2 && peak <= 30.8)
    {
      digitalWrite(LED_B, HIGH);
    }
    else
    {
      digitalWrite(LED_B, LOW);
    }
}
```

A-6 Complete System Test

```
#include "arduinoFFT.h"

#define SAMPLES 256             //Must be a power of 2
#define SAMPLING_FREQUENCY 250 //Hz, must be less than 10000 due to
ADC

arduinoFFT FFT = arduinoFFT();

unsigned int sampling_period_us;
unsigned long microseconds;

double vReal[SAMPLES];
double vImag[SAMPLES];

int speakerPin = 12;
int speakerState = LOW;
unsigned long previousMillis = 0;
const long interval = 100;

float binNum;
float delta = 0; float theta = 0; float alpha = 0; float beta = 0;
float other = 0;
float deltaR; float thetaR; float alphaR; float betaR; float otherR;
float deltaP; float thetaP; float alphaP; float betaP; float otherP;
float totalPwr = 0; float totalPwrR;

int readValue;
float voltage;
float out_data;

#define LED_A 22
#define LED_B 24

void setup() {
    //System external reference voltage set to 0.825V
    analogReference(EXTERNAL);
    Serial.begin(500000);

    //Set pins corresponding to negative/positive feedback to be out-
puts
    pinMode(LED_A, OUTPUT);
    pinMode(LED_B, OUTPUT);

    //Calculate sampling period in microseconds
    sampling_period_us = round(1000000*(1.0/SAMPLING_FREQUENCY));
}

void loop() {
    /*SAMPLING*/
    for(int i=0; i<SAMPLES; i++)
    {
```

```
        microseconds = micros();    //Overflows after around 70 min-
utes!

        sample();                   //Calls the sample function

        vReal[i] = out_data;        //Filtered data passed into FFT
        vImag[i] = 0;

        while(micros() < (microseconds + sampling_period_us)){}
    }

    /*FFT*/
    FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
    FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
    FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);

    //Find the signal's peak frequency from FFT analysis
    double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING_FREQUENCY);


    for(int i=0; i<(SAMPLES/2); i++)
    {
        binNum = (i * 1.0 * SAMPLING_FREQUENCY) / SAMPLES;

        //Sum the power of brain waves in each frequency range
        if(binNum > 1 && binNum <= 4) {delta += sq(abs(vReal[i]));}
        if(binNum > 4 && binNum <= 8) {theta += sq(abs(vReal[i]));}
        if(binNum > 8 && binNum <= 13) {alpha += sq(abs(vReal[i]));}
        if(binNum > 13 && binNum <= 30.5) {beta +=
sq(abs(vReal[i]));}
        if(binNum > 30.5 && binNum <= (SAMPLING_FREQUENCY/2)) {other
+= sq(abs(vReal[i]));}
        if(binNum > 1 && binNum <= (SAMPLING_FREQUENCY/2)) {totalPwr
+= sq(abs(vReal[i]));}
        else{}
    }

    //Calculate the percent of total power of each frequency range
    deltaP = findPercentSignalPwr(delta,totalPwr);
    thetaP = findPercentSignalPwr(theta,totalPwr);
    alphaP = findPercentSignalPwr(alpha,totalPwr);
    betaP = findPercentSignalPwr(beta,totalPwr);
    otherP = findPercentSignalPwr(other,totalPwr);

    //Calculate the relative signal power of each frequency range in
dB
    deltaR = findRelativeSignalPwr(delta,totalPwr);
    thetaR = findRelativeSignalPwr(theta,totalPwr);
    alphaR = findRelativeSignalPwr(alpha,totalPwr);
    betaR = findRelativeSignalPwr(beta,totalPwr);
    otherR = findRelativeSignalPwr(other,totalPwr);
```

```
    //Store calculated values in relative and percent signal power
arrays
    float sigPwrR[5] = {deltaR, thetaR, alphaR, betaR, otherR};
    float sigPwrP[5] = {deltaP, thetaP, alphaP, betaP, otherP};

    //Create array of signal names corresponding to order calcula-
tions were stored
    String sigNames[5] = {"delta", "theta", "alpha", "beta",
"other"};

    //Name the formats to be used in the funtion to print results
    String formats[2] = {"RELATIVE", "PERCENT"};
    float* sigPwr[2] = {sigPwrR, sigPwrP};

 /*NOTICE: IT IS RECOMMENDED TO USE ONLY ONE RESULTS OUTPUT AT A
TIME, EITHER 'PRINT RESULTS'
   OR 'LED & Speaker Feedback'. CHOOSE A SECTION, THEN COMMENT THE
OTHER ONE OUT BEFORE RUNNING*/

    /*PRINT RESULTS*/
    Serial.println("Peak = " + String(peak));     //Print out what
frequency is the most dominant.

    //Find frequency range containing the highest percent power
    int range = findMaxSignalRange(sigPwrP);

    //This loop feeds all calculated results through a print function
for display
    for(int j = 0; j < 2; j++)
    {
      Serial.println(formats[j] + " SIGNAL POWER:");

      for(int i = 0; i < 5; i++)
      {
          printSignalPwr(sigNames[i],sigPwr[j][i],formats[j]);
      }
      Serial.println();
    }


    /*LED & Speaker Feedback:
    This demo is set for distinguishing between eyes open and eyes
closed, where eyes open
    corresponds to beta (13-30Hz) and eyes closed corresponds to in
alpha (8-13Hz).*/
    unsigned long currentMillis = millis();

    //If the max signal power falls in alpha, provide negative feed-
back.
    if(range == 2)
    {
      //Turn on the red LED.
```

```
      digitalWrite(LED_A, HIGH);

      //Turn on buzzer to beep periodically, as an alarm
      if(currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;
        if(speakerState == LOW)
        {
          tone(speakerPin, 261);
          speakerState = HIGH;
        }
        else
        {
          noTone(speakerPin);
          speakerState = LOW;
        }
      }
    }
    else
    {
      digitalWrite(LED_A, LOW);
      speakerState = LOW;
      noTone(speakerPin);
    }

    //If the max signal power falls in beta, provide positive feed-
back.
    if(range == 3)
    {
      //Turn on the green LED
      digitalWrite(LED_B, HIGH);
    }
    else
    {
      digitalWrite(LED_B, LOW);
    }
}

//Takes an analog reading and passes it through fir_filt
void sample()
{
  //Read in analog signal, using channel 1
  readValue = analogRead(1);

  //Convert analogRead to voltage, shift signal for DC bias correc-
tion, scale by 10 for visualization
  voltage = 10*((readValue * (0.825/1024)-0.25));

  //Pass input through LPF: can use analog reading directly or calcu-
lated voltage
  fir_filt((float)voltage, &out_data);
}
```

```
//Calculates the relative signal power for each brainwave frequency
range in dB
float findRelativeSignalPwr(float pwr, float base)
{
  return 10*log10(pwr/base);
}

//Calculates the percentage of total signal power for each brainwave
frequency range
float findPercentSignalPwr(float pwr, float base)
{
  return (pwr/base)*100;
}

//Print function: Displays the power corresponding to each brainwave
frequency range
void printSignalPwr(String sigName, float relativePwr, String format)
{
  String a;

  sigName.toLowerCase();
  format.toLowerCase();

  if(format == "relative") {a = "dB";}
  else {a = "%";}

  if(sigName == "delta") {Serial.println("Delta = " +
String(relativePwr) + a);}
  if(sigName == "theta") {Serial.println("Theta = " +
String(relativePwr) + a);}
  if(sigName == "alpha") {Serial.println("Alpha = " +
String(relativePwr) + a);}
  if(sigName == "beta") {Serial.println("Beta = " +
String(relativePwr) + a);}
  if(sigName == "other") {Serial.println("Other = " +
String(relativePwr) + a);}
  else{}
}

//Finds the frequency range containing the greatest power. Returned
as an array index.
int findMaxSignalRange(float sigPwrR[5])
{
  //Find max signal range
  int index = 0;
  float maxPwr = sigPwrR[index];

  for(int i = 1; i < 5; i++)
  {
    if(sigPwrR[i] > maxPwr)
    {
      maxPwr = sigPwrR[i];
```

```
      index = i;
      //Indicies correspond to delta (0), theta (1), alpha (2), beta
(3), & other (4)
    }
  }

  /* OPTIONAL: Print the max signal range name
  switch (index) {
  case 0:
    Serial.println("DELTA");
    break;
  case 1:
    Serial.println("THETA");
    break;
  case 2:
    Serial.println("ALPHA");
    break;
  case 3:
    Serial.println("BETA");
    break;
  default:
    Serial.println("OTHER");
  }
  */
  return index;
}

//Low pass filter function: convolves input with filter coefficients
void fir_filt(float in_data, float* out_data)
{
  static const int N = 21; //Filter length.

  //Filter coefficients.  Only half will be used as the coefficients
are symmetric.
  static const float coeff[] =
  {
    0.001214735329243, -0.012960952370491, -0.010093833713808,
 0.013957771298692,  0.031889096998368,
    0.000124591983717, -0.062115960863680, -0.056006130074433,
 0.089034472964041,  0.299161329745819,
    0.400142978571672,  0.299161329745819,  0.089034472964041, -
0.056006130074433, -0.062115960863680,
    0.000124591983717,  0.031889096998368,  0.013957771298692, -
0.010093833713808, -0.012960952370491,
    0.001214735329243
  };

  static float shift_reg[N]; //Shift register for storing incomming
data values.

  float acc = 0; //Accumulator.
```

```
  //Shift all values.
  for (int i = N - 1; i >= 0; i--)
  {
    shift_reg[i+1] = shift_reg[i];
  }

  shift_reg[0] = in_data; //Get input data.

  //Multiply through. Take advantage of symmetry.
  for (int i = 0; i < N/2; i++)
  {
    acc += coeff[i] * (shift_reg[i] + shift_reg[N - 1 - i]);
  }

  //Need the middle value if there is an odd number of filter taps.
  if(N % 2) acc += shift_reg[N / 2] * coeff[N / 2];

  *out_data = acc; //Output result.
}
```